

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**NESNE KAVRAMA BECERİLERİNİN DERİN PEKİŞTİRMELİ
ÖĞRENME ALGORİTMALARIYLA GELİŞTİRİLMESİ**

Musab COŞKUN

Doktora Tezi

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Elektronik Bilim Dalı

ARALIK 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Doktora Tezi

NESNE KAVRAMA BECERİLERİNİN DERİN PEKİŞTİRMELİ ÖĞRENME ALGORİTMALARIYLA GELİŞTİRİLMESİ

Tez Yazarı
Musab COŞKUN

Danışman
Prof. Dr. Yakup DEMİR

İkinci Danışman
Doç. Dr. Özal YILDIRIM

ARALIK 2022
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı:	Nesne Kavrama Becerilerinin Derin Pekiştirmeli Öğrenme Algoritmalarıyla Geliştirilmesi
Yazarı:	Musab COŞKUN
İlk Teslim Tarihi:	07.11.2022
Savunma Tarihi:	07.12.2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Prof. Dr. Yakup DEMİR Fırat Üniversitesi, Mühendislik Fakültesi	<i>İmza</i> Onayladım
Başkan:	Doç. Dr. Ahmet ÇINAR Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Doç. Dr. Nuh ALPASLAN Bingöl Üniversitesi, Mühendislik Mimarlık Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Sencer ÜNAL Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Ali ARSERİM Dicle Üniversitesi, Mühendislik Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Burhan ERGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “ Nesne Kavrama Becerilerinin Derin Pekiştirmeli Öğrenme Algoritmalarıyla Geliştirilmesi ” Başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

07.12.2022

Musab COŞKUN



ÖNSÖZ

Pekiştirmeli öğrenme modellerinin eğitilmesi ve kurgulanması kolay değildir. Bir görüntü işleme modelinde birkaç milyon resme ihtiyaç duyulurken pekiştirmeli öğrenme modelinde bu sayı on milyarlar yüz milyarlar kadar gidebilmektedir. Böyle bir veriyi gerçek hayatta toplamak çok zor olmaktadır. O yüzden simülasyon ortamları kullanılmaya başlanmıştır. Simülasyon ortamlarında, gerçek hayatın bir versiyonu sanal ortamda oluşturulmaktadır. Robotik simülatörleri sayesinde yürüme, kavrama gibi robotik problemler hızlı bir şekilde incelenebilmektedir. Bu tez çalışmasında bir robot manipülatörüyle nesne kavrama görevinin gerçekleştirilmesi amaçlanmıştır. Bu doğrultuda kavrama hareketlerinin anlaşılması için sEMG yöntemi ile toplanan kavrama sinyallerinden oluşan bir veri kümesi üzerindeki sinyalleri sınıflandıran bir derin sinir ağı önerilmiştir. Sonraki aşamada derin pekiştirmeli öğrenme yöntemi ile Pong oyununu oynayabilen bir uygulama gerçekleştirilmiştir. Son olarak simülasyon ortamında robotik bir kavrayıcı ile kavrama eyleminin nasıl gerçekleştirilebileceği ve derin pekiştirmeli öğrenmenin performansı incelenmiştir. Böylece bu çalışma ile robotikte önemli bir problem olan kavrama eyleminin gerçekleştirilmesi ve ileriye yönelik yeni çalışmaların yapılması için teknolojik bir zemin hazırlanmış olup bilime katkı sağlanmaktadır. Ayrıca, bu tez çalışması, ilgimi çeken ve çalışmaktan keyif aldığım konulardan biri olan derin öğrenme yöntemleri ile önümüzdeki süreçte yapacağım çalışmaları gerçekleştirmenin ilk adımıdır.

Çalışmalarımın gerçekleşmesi için gerekli imkânları sağlayan, bilgilerini ve yardımlarını eksik etmeyen ve akademik hayatım boyunca her zaman desteğini hissettiğim çok kıymetli danışman hocam Sayın Prof. Dr. Yakup DEMİR'e en içten teşekkürlerimi sunarım. Zaman ve mekân gözetmeksizin yardımlarını hiçbir zaman esirgemeyen, değerli bilgileri ve deneyimleriyle tez çalışmama önemli katkılar sağlayan ikinci danışman hocam Doç. Dr. Özal YILDIRIM'a teşekkürü bir borç bilirim.

Bugünlere gelmemde emekleri olan çok sevdiğim kıymetli anneme ve babama sonsuz şükranlarımı sunarım. Hayatım boyunca desteklerini her zaman hissettiğim, moral ve motivasyonlarıyla hep yanımda olan sevgili kardeşlerim Zeynep, Fatima ve Hüseyin'e çok teşekkür ederim. Ayrıca manevi desteğiyle hep yanımda olan değerli eşim Betül COŞKUN'a ve çalışmalarım boyunca yardımcı olan tüm arkadaşlarıma teşekkürlerimi sunarım.

Musab COŞKUN
ELAZIĞ, 2022

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xii
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
1.1. Genel Bakış	1
1.2. Literatür Özeti	2
1.3. Tezin Amacı	3
1.4. Tezin Yapısı.....	3
2. ROBOTİĞE GENEL BİR BAKIŞ.....	5
2.1. Robot Kinematiği	8
2.2. Robot Dinamiği	8
2.3. Robot Mekanizması ve Tahriki	9
2.4. Robotik Algılama ve Tahmin	9
2.5. Robotikte Hareket Planlaması	10
2.6. Robotikte Hareket Kontrolü	11
2.7. Robotikte Kuvvet Kontrolü	12
2.8. Robotikte Sistem Mimarileri ve Programlama	12
2.9. Robotikte Yapay Zekâ Akıl Yürütme Yöntemleri.....	13
2.9.1. Bilgi Temsili ve Çıkarım	13
2.9.2. Robotikte Eylem Planlama	14
2.9.3. Robotlarda Öğrenme	15
2.10. Robotik Uygulama Alanları.....	16
2.10.1. Endüstriyel Robotik.....	17
2.10.2. Sualtı Robotiği	19
2.10.3. Hava Robotiği	20
2.10.4. Uzay Robotları Ve Sistemleri.....	21
2.10.5. Tarım ve Ormancılıkta Robotik	22
2.10.6. İnşaatda Robotik	22
2.10.7. Maden Robotiği.....	23
2.10.8. Tıbbi Robotik	23
2.10.9. Rehabilitasyon ve Sağlık Hizmetleri Robotiği	24
2.11. Robotikte Karşılaşılan Temel Problemler.....	25
2.11.1. Yürüme.....	25
2.11.2. Kavrama	26
2.11.3. Kavrayıcıların Sınıflandırılması	27
2.11.4. Robotik Kavrama	29
2.11.5. Kavrama Yöntemleri	30
2.11.6. Sensör Tabanlı Kontrol Stratejileri.....	31

3. DERİN PEKİŞTİRMELİ ÖĞRENME	35
3.1. Makine Öğrenmesine Genel Bir Bakış	35
3.2. Derin Öğrenmenin Temelleri.....	38
3.3. Yapay Sinir Ağları.....	39
3.3.1. Çok Katmanlı Algılayıcılar	41
3.3.2. Yaygın Aktivasyon Fonksiyonları.....	42
3.3.3. Sinir Ağlarında İleri ve Geri Yayılım.....	45
3.4. Derin Öğrenme Yöntemleri	46
3.4.1. Denetimsiz Öğrenmede Derin Öğrenme	46
3.4.2. Denetimli Öğrenmede Derin Öğrenme	53
3.4.3. Derin Öğrenmede Kullanılan Hiperparametrelerin Belirlenmesi	63
3.4.4. Derin Öğrenmede Optimizasyon Yöntemleri.....	65
3.4.5. Derin Öğrenme Modellerinin Performans Ölçütleri.....	66
3.4.6. Derin Öğrenme Yazılım Araçları	68
3.5. Derin Pekıştirmeli Öğrenmeye Genel Bir Bakış.....	70
3.5.1. Markov Karar Süreçleri.....	72
3.5.2. Bellman Denklemi.....	75
3.5.3. Pekıştirmeli Öğrenme Terminolojisi	76
3.5.4. Q-Öğrenme Algoritması.....	78
3.5.5. Derin Pekıştirmeli Öğrenme.....	79
3.5.6. DPÖ Yöntemleri.....	81
4. UYGULAMALAR.....	91
4.1. yEMG Verileri Kullanılarak 1B-KSA Modeli ile Kavrama Uygulaması	91
4.1.1. İlgili Literatür	91
4.1.2. Veri Kümesi	93
4.1.3. Önışleme Aşaması.....	95
4.1.4. Önerilen KSA Modelinin Yapısı	95
4.1.5. Uygulamalar	97
4.1.6. Tartışma.....	104
4.1.7. Sonuçlar.....	107
4.2. DKQÖ Algoritması ile Pong Oyunu.....	107
4.2.1. İlgili Literatür	108
4.2.2. Önerilen Yöntem	110
4.2.3. Uygulama Sonuçları	113
4.2.4. Sonuçlar.....	115
4.3. Düello DQA Algoritması ile Robotik Kavrama Uygulaması	115
4.3.1. İlgili Literatür	116
4.3.2. Uygulama Organizasyonu	117
4.3.3. Uygulama Sonuçları	128
5. SONUÇLAR.....	131
ÖNERİLER VE GELECEK ÇALIŞMALAR.....	132
KAYNAKLAR	133
ÖZGEÇMİŞ	

ÖZET

Nesne Kavrama Becerilerinin Derin Pekiştirmeli Öğrenme Algoritmalarıyla Geliştirilmesi

Musab COŞKUN

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Aralık 2022, Sayfa: xiii + 148

Yapay zekânın farklı alanlarda gelişim göstermesiyle robotların farklı uygulama alanlarında kullanımının artması, bulundukları ortamla olan etkileşim ihtiyacını da arttırmıştır. Bu amaçla son zamanlarda robot uç işlevcileri ortamla etkileşimi sağlarken dokunma, görme gibi çeşitli algılama bileşenleri kullanılmaya başlanmıştır. Bir robotik uç işlevcisinin eklemlerine ait parametrelerin göz önünde bulundurulmadan ortam değişikliklerinin kamera tarafından algılanabilmesi, kamera gibi görsel algılama araçlarını faydalı bir araç haline getirmiştir. Görme tabanlı bir sistemde, imgelerden alınan ortam özellikleri ile kavrama gibi verilen bir görevi yerine getirmek için robot kolunun yapması gereken hareketlerin belirlenmesi önem arz etmektedir.

Bu tezde Webots simülatör ortamında kavrama senaryosu oluşturulmuştur. Kameradan alınan görüntülere bağlı olarak 5 serbestlik dereceli KUKA youBot robot kolunun kavrama performansının ölçülmesinin hedeflendiği temel bir robotik uygulama çerçevesi sunulmuştur. Kavrama eylemi derin pekiştirmeli öğrenmeye dayalı bir öğrenme stratejisi ile geliştirilmiştir. Bu bağlamda ilk olarak yüzey ElektroMiyoGram (yEMG) sinyallerinden oluşan insandaki kavrama tiplerinin otomatik olarak sınıflandırılmasında tek boyutlu Konvolüsyonel Sinir Ağı (KSA) yapısı kullanılmıştır. Daha sonra Pong oyununu oynayabilen bir Derin Konvolüsyonel Q Ağı (DKQA) modeli geliştirilmiştir. Geliştirilen bu model esas alınarak KUKA youBot robot kolu ile Webots simülasyon ortamında kavramayı öğrenebilen temel bir öğrenme yapısı olan düello DKQA yöntemi başarılı bir şekilde oluşturulmuştur. Ardından simülasyon ortamında yapay zeka destekli, geliştirilebilir bir kavrama sisteminin modeli elde edilmiştir.

Bu tez çalışması ile birçok uygulamada karşılaşılan nesne kavrama becerileri geliştirilmiştir ve bu doğrultuda derin öğrenme ve pekiştirmeli öğrenme temelli farklı algoritmalar sunulmuştur. Tezde endüstrideki mevcut problemler belirlendikten sonra nesne kavramaya yönelik sınıflandırma ve tahmin becerileri geliştirilmiştir. Özellikle endüstriyel alanın bazı süreçlerinde kullanıcıların problemlerine çözüm önerileri sunulmuştur.

Anahtar Kelimeler: Nesne Kavrama, Derin Öğrenme, Derin Pekiştirmeli Öğrenme

ABSTRACT

Developing Object Grasping Skills with Deep Reinforcement Learning Algorithms

Musab COŞKUN

Ph.D. Thesis

FIRAT UNIVERSITY

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

December 2022, Pages: xiii + 148

With the development of Artificial Intelligence in different areas, the increase in the use of robots in different application areas has also increased the need for interaction with their environment. For this purpose, various sensing components such as touch and vision have been used recently, while robot end processors provide interaction with the environment. The ability to detect changes in the environment by the camera without considering the parameters of the joints of a robotic end effector has made visual detection tools such as a camera a very useful tool. In a vision-based system, it is important to determine the movements of the robot arm to fulfill a given task such as grasping with the environment properties taken from the images.

In this thesis, a comprehension scenario was created in the Webots simulator environment. Based on the images taken from the camera, a basic robotics application framework is presented, in which it is aimed to measure the grip performance of the KUKA youBot robot arm with 5 degrees of freedom. The act of grasping was developed with a learning strategy based on deep reinforcement learning. In this context, firstly, a one-dimensional deep Convolutional Neural Network (CNN) structure was used in the automatic classification of grasp types in humans consisting of Surface Electromyogram signals. Later, a Deep Convolutional Q Network (DCQN) model was developed that can play the Pong game. Based on this developed model, a duelling DCQN method, which is a basic learning structure that can learn to grasp in the Webots simulation environment, has been successfully created with the KUKA youBot robot arm. As a result, a model of a developable grasping system supported by artificial intelligence was obtained in the simulation environment.

With this thesis, object grasping skills encountered in many applications have been developed and in this direction, different algorithms based on deep learning and reinforcement learning are presented. In the thesis, after identifying the existing problems in the industry, classification and estimation skills for object grasping were developed. Especially in some processes of the industrial area, solutions to the problems of the users are presented.

Keywords: Object grasping, Deep Learning, Deep Reinforcement Learning

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. General Motors fabrikasında çalışan ilk Unimation robotu [25]	6
Şekil 2.2. Robotun basitleştirilmiş halinin şematik gösterimi	8
Şekil 2.3. Unimation şirketi tarafından geliştirilen ilk endüstriyel robotun (a) patenti ve (b) kendisi[15]... 17	
Şekil 2.4. (a) Araba gövdesi montajı adımları ve (b) üretim hattında görev yapan montaj robotları.....	19
Şekil 2.5. Boyama robotları. (a) Araba gövdesi boyama için çok robotlu bir çalışma hücresi (b) Yüksek hızlı dönen boya tabancası	19
Şekil 2.6. (a) Bir telerobotik hizmet sağlayıcının kavramsal tasarımı [67], (b) uzay mekiği uzaktan manipülatör sistemi [68]	21
Şekil 2.7. Eklem rekonstrüksiyon cerrahisi için klinik olarak uygulanan ilk robot [82,83].	24
Şekil 2.8. İlk insan benzeri modern robot [91]	25
Şekil 2.9. Farklı sınıflandırma stratejilerine göre farklı robotik kavrayıcılar [118].....	28
Şekil 3.1. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki.....	36
Şekil 3.2. Sabit kodlama ile makine öğrenmesi arasında fark	37
Şekil 3.3. KSA tarafından algılanan yüz özellikleri hiyerarşisi örneği [172,173]	39
Şekil 3.4. 8 girişli ve 4 çıkışlı basit bir sinir ağı ve DSA'nın gösterimi [174].....	39
Şekil 3.5. Bir nöronun-sinir hücresinin biyolojik gösterimi	40
Şekil 3.6. Algılayıcı(Perceptron) matematiksel modeli	41
Şekil 3.7. Basit bir algılayıcı çıkışı	42
Şekil 3.8. Farklı yaygın aktivasyon fonksiyonları olan sigmoid fonksiyonu (a), tanh fonksiyonu (b) ve ReLU fonksiyonuna (c) ait grafiksel gösterimler [181]	43
Şekil 3.9. Yaygın ReLU aktivasyon fonksiyonu tiplerine ait grafiksel gösterimler	44
Şekil 3.10. Dört görünür katmanlı ve üç gizli katmanlı kısıtlanmamış Boltzmann Makinesi mimarisi	46
Şekil 3.11. KBM Mimarisi	47
Şekil 3.12. Üç katmanlı KBM'den oluşan geleneksel bir DİA mimarisi	49
Şekil 3.13. Otomatik kodlayıcı örneği [203]	49
Şekil 3.14. Otomatik kodlayıcı gösterimi	50
Şekil 3.15. Gizli katmanın giriş katmanı olarak alınması ve tekrar oluşturulması gösterimi	51
Şekil 3.16. İki katmanlı yığın otomatik kodlayıcı örneği	51
Şekil 3.17. Klasik bir ÇÜA yapısı [152,206]	53
Şekil 3.18. ÇÜA ile elde edilen farklı yüzler [208].....	53
Şekil 3.19. Konvolüsyonel ve örnekleme katmanlarından oluşturulmuş LeNet mimarisi [148]	55
Şekil 3.20. Basit bir KSA yapısı örneği [148]	56

Şekil 3.21. (a) 8 imgesi ve (b, c) piksel değerleri ile temsili [213].....	56
Şekil 3.22. İmge, kernel ve özellik haritasının kullanıldığı konvolüsyonel süreç	57
Şekil 3.23. 3 farklı derinlikli özellik haritası	58
Şekil 3.24. Maksimum örnekleme örneği.....	58
Şekil 3.25. Basit bir ÖSA örneği [147]	60
Şekil 3.26. Bir ÖSA'nın ileriye dönük hesaplamasında yer alan hesaplama zamanında açılması	60
Şekil 3.27. Orijinal UKSBA mimarisi [214]	61
Şekil 3.28. Kapı Özyinelemeli Birim hücre mimarisi ve bağlantıları [219]	63
Şekil 3.29. (a) İzgara aramasının (b) rastgele aramayla karşılaştırılması [223]	64
Şekil 3.30. Pekiştirmeli öğrenme süreci [243].....	71
Şekil 3.31. Cartpole sistemi [244]	72
Şekil 3.32. Temel bir Q-öğrenme sürecinin adımları	78
Şekil 3.33. Derin öğrenme ile pekiştirmeli öğrenmenin birleştirilmesi	80
Şekil 3.34. DQÖ algoritması süreci.....	84
Şekil 3.35. DQÖ yöntemine ait bir gösterim [255]	85
Şekil 3.36. Toplama katmanı içeren bir düello DQA mimarisi [257]	89
Şekil 4.1. Kavrama tipleri.....	94
Şekil 4.2. Tüm kavrama tiplerine ait örnek sinyaller.....	94
Şekil 4.3. Ham bir EMG sinyali ve medyan filtre uygulanmış bir EMG sinyali örnekleri.....	95
Şekil 4.4. Önerilen 11 katmanlı KSA modelinin blok diyagramı	96
Şekil 4.5. 1. veri kümesindeki kanal-1 (a, b) ve kanal-2'ye (c, d) ait veriler üzerinde yapılan çalışmalardan elde edilen sonuçlar	98
Şekil 4.6. 2. veri kümesine ait kanal-1 (a, b) ve kanal-2 (c, d) verileri üzerinde yapılan çalışmalardan elde edilen sonuçlar	99
Şekil 4.7. Karışık veri kümesine ait kanal-1 (a, b) ve kanal-2 (c, d) verileri üzerinde yapılan çalışmalardan elde edilen sonuçlar	100
Şekil 4.8. (a) 2. veri kümesine ait kanal-1, (b) 1. veri kümesine ait kanal-2 ve (c) karışık veri kümesine ait kanal-2 için elde edilen hata matrisleri	101
Şekil 4.9. Bir bulut sisteminde eğitilen derin öğrenme modeli.....	106
Şekil 4.10. Bir pekiştirmeli öğrenme ajanının çevre ile etkileşimi [20]	111
Şekil 4.11. Q-öğrenme ve DQÖ'nün karşılaştırılması.....	111
Şekil 4.12. DKQÖ ile pekiştirmeli öğrenme süreci [249]	112
Şekil 4.13. Pong oyunu uygulaması için önerilen KSA yapısı.....	113
Şekil 4.14. Pygame kullanılarak oluşturulan Pong Oyunu ekranı	113
Şekil 4.15. Eğitim aşamasında elde edilen skor değerinin değişimi	115

Şekil 4.16. Uygulama sonunda elde edilen ödül değerinin değişimi	115
Şekil 4.17. Örnek MuJoCo Ortamı	118
Şekil 4.18. Örnek Gazebo ortamı	119
Şekil 4.19. Örnek Gazebo ortamı	119
Şekil 4.20. Webots ortamı	120
Şekil 4.21. Simülasyonda kullanılan 5 serbestlik dereceli KUKA youBot robot kolu yapısı.....	123
Şekil 4.22. KUKA youBot robot kolunun yükseklik ölçüleri ve açı sınırları	123
Şekil 4.23. Deney Düzeneği	124
Şekil 4.24. Kinect Kamera yapısı	125
Şekil 4.25. Önerilen pekiştirmeli öğrenmeye dayalı robotik kavrama yapısı	127
Şekil 4.26. Düello DKQA yapısında kullanılan adımlar	127
Şekil 4.27. Eylem seçimi kod parçacığı	129
Şekil 4.28. Anlık ödül değeri değişimi	130
Şekil 4.29. Anlık ödül değeri değişimi	130

TABLÖLAR LİSTESİ

Sayfa

Tablo 3.1. Karmaşıklık matrisi tablosu	67
Tablo 4.1. Önerilen modelin parametreleri	96
Tablo 4.2. Doğrulama seti olarak 1. veri kümesinin kanal-1'i kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans	101
Tablo 4.3. Doğrulama seti olarak 1. veri kümesinin kanal-2'si kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans	102
Tablo 4.4. Doğrulama seti olarak 2. veri kümesinin kanal-1'i kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans	102
Tablo 4.5. Doğrulama seti olarak 2. veri kümesinin kanal-2'si kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans	103
Tablo 4.6. Doğrulama seti olarak karışık veri kümesinin kanal-1'i kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans	103
Tablo 4.7. Doğrulama seti olarak karışık veri kümesinin kanal-2'si kullanılarak 5-fold çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans	104
Tablo 4.8. Farklı çalışmaların performansları ve özellikleri karşılaştırılması	105
Tablo 4.9. DQÖ algoritmasının sözde kodu	112
Tablo 4.10. DKQÖ algoritmasının eğitim aşamasında kullanılan parametreler ve hiperparametreler	114
Tablo 4.11. Düello DQA Algoritması Hiperparametre Ayarları	128

SİMGELER VE KISALTMALAR

Simgeler

G_t	: Getiri
s_t	: t anında ajanın durumu
a_t	: t anında ajanın gerçekleştirdiği aksiyon
r_t	: t anında ajanın aldığı ödül değeri
s_{t+1}	: t anından sonraki ajanın durumu

Kısaltmalar

KSA	: Konvolüsyonel Sinir Ağları
DQA	: Derin Q Ağı
DKQA	: Derin Konvolüsyonel Q Ağı
DKQÖ	: Derin Konvolüsyonel Q Öğrenme
DSA	: Derin Sinir Ağları
BSA	: Biyolojik Sinir Ağı
YSA	: Yapay Sinir Ağı
DOF	: Serbestlik Derecesi (Degrees of Freedom)
ReLU	: Rectified Linear Unit
KBM	: Kısıtlı Boltzmann Makinesi
DİA	: Derin İnanç Ağı
ÇÜA	: Çekişmeli Üretici Ağlar
ÖSA	: Özyinelemeli Sinir Ağları
UKSBA	: Uzun Kısa Vadeli Bellek Ağları
LSTM	: Long Short Term Memory
MKS	: Markov Karar Süreci
DPÖ	: Derin Pekiştirmeli Öğrenme
İME	: İnsan Makine Etkileşimi
yEMG	: Yüzey ElektroMiyogram
EKG	: ElektroKardiyoGrafı
EEG	: ElektroEnsefaloGrafı
EOG	: ElektroOküloGrafı
DDİ	: Doğal Dil İşleme

1. GİRİŞ

1.1. Genel Bakış

Yetenekli ve zeki makineler yaratma hayali, insanlık tarihinde sürekli var olmuştur. İnsanoğlunun kendisini Homo Sapiens (bilge insan) olarak görmesinin nedeni zekânın öneminden kaynaklanmaktadır. Nasıl düşündüğümüzü anlamaya çalışmakla birlikte benzer şekilde düşünen makineler icat etme çabaları da süregelmektedir. Yapay zekâ alanı, karmaşık bir dünyayı sadece anlamaya değil, aynı zamanda akıllı varlıklar inşa etmeye de çalışmaktadır. Yapay zekâ, bilim ve mühendislikteki en önemli alanlardan biridir. Yapay zekâ ile ilgili çalışmalar, İkinci Dünya Savaşı'ndan kısa süre sonra başladı ve sonrasında 1956 yılında yapay zekâ teriminin kendisi ortaya atılmıştır. Yapay zekâ, evrensel bir alan olup zekâ gerektiren herhangi bir görevle ilgilenir. Yapay zekâ, robotla bir nesneyi kavrama, Go oyunu oynama, matematiksel teoremleri kanıtlama, şiir yazma, kalabalık bir caddede araba kullanma ve hastalıkları teşhis etme gibi çok çeşitli alt alanları kapsamaktadır.

Günümüzde robotlar, endüstriyel üretimden sağlık hizmetlerine, ulaşımdan uzay ve denizlerdeki keşiflere kadar modern yaşamın birçok alanında önemli bir etki yaratmaktadır. Gelecekte robotların günümüzdeki kişisel bilgisayarlar kadar yaygın ve kişiye özgü olabilecek duruma gelmesi öngörülmektedir. Robotlar yaşlı bakımı, tarım işleri, afet müdahalesi ve savaş alanı operasyonları gibi birçok görevde insanlara yardımcı olmak için faydalanılmaktadır. Günümüzde ve gelecekteki tüm bu uygulamalarda, robotlar kendilerine atanan görev hedeflerini gerçekleştirmek için bulundukları ortamlardaki nesneleri kavrama ve manipüle etme yeteneklerine sahip olmalıdır. Bu tür robotik işlemler, görevle ilgili nesneleri güvenli bir şekilde kavramak ve görev hedeflerini gerçekleştirmek için robotik kavrayıcıların (tutucular/ellerin) kullanımını gerektirmektedir. Nasıl insan eli insan manipülasyonunun organıysa, robotik kavrayıcılar da genellikle ortamda bulunan nesnelerle doğrudan temas halinde olan robot parçalarıdır.

Bu sebeple bu tez çalışmasında birçok alanda ihtiyaç duyulan ve özellikle robotların ortamla etkileşime geçmesini sağlayan uç işlevciler olan robotik bir kavrayıcı ile nesne kavrama görevini öğrenebilen bir yapay zekâ yönteminin geliştirilmesi hedeflenmiştir. Bu bağlamda güncel yapay zekâ yaklaşımlarından biri olan Derin Pekiştirmeli Öğrenme (DPÖ) teknikleri incelenmiştir. DPÖ etkin bir şekilde kullanılmasının sebebi derin öğrenmede yaşanan devrim niteliğindeki yaklaşımlar olmuştur. Derin öğrenme, yapay zekâ biliminin altında ele alınan araştırma alanlarından biridir [1]. Derin Sinir Ağları (DSA); veri ile öğrenebilme, doğrusal olmamaları ve genelleme yapabilmeleri özellikleri sayesinde esnek ve güçlü araçlar olarak ön plana çıkmaktadır [2]. Bu sayede görüntü işleme, örüntü tanıma ve robotik gibi alanlarda oldukça zor problemleri çözebilecek yeteneklere sahip yeni yaklaşımlar geliştirilmesinin önü açılmıştır.

1.2. Literatür Özeti

Kavrama, robotikte karşılaşılan kritik problemlerden biridir [3]. İnsanlarla aynı el becerisi seviyesinde çalışabilen bilişsel robotların yapılması, araştırma ve endüstrideki yoğun çalışmalara rağmen çözülmemiş bir sorun olmaya devam etmektedir [4,5]. Dinamik ve yapılandırılmamış (organize edilmemiş) ortamlarda çalışabilen robotların geliştirilmesi büyük ilgi görmektedir. Robotik kavrama yaklaşımları, verilen bir görevi herhangi bir insan müdahalesi olmadan otomatik olarak yapılandırmak için yazılımsal yoğunluğu önemli ölçüde azaltan öğrenmeye dayalı yöntemler kullanır [6]. Makine öğrenmesi, yeni nesnelere genelleme yeteneği nedeniyle robotik kavrama için gelecek vadeden bir yaklaşımdır [3].

Görsel sensör tabanlı robotik kavramaya yönelik yaklaşımlar, birçok farklı ölçüte göre kategorize edilebilir. Genel olarak bu yaklaşımlar analitik ve veriye dayalı yöntemler şeklinde ayrılabilir [7,8]. Analitik yaklaşımlar, uygun bir kavrama pozunu belirlemek için tipik olarak hedef nesnenin şeklini analiz eder. Veriye dayalı (empirik) yaklaşımlar, makine öğrenimine dayalı olup son yıllarda popülerlik kazanmıştır. Artan veri kullanılabilirliği, daha iyi hesaplama kaynakları ve algoritmik gelişmeler veriye dayalı yaklaşımlarda önemli gelişmelerin kaydedilmesini sağlamıştır [3]. Ayrıca görsel sensör tabanlı yaklaşımlar, verilen görevi yerine getirmek için nesneye ilişkin belirli bilgilerin kullanılıp kullanılmadığına bağlı olarak model tabanlı veya modelden bağımsız şeklinde de kategorize edilebilir [9]. Katı nesneler için model tabanlı yaklaşımlar tipik olarak bir poz tahmini adımı içerir ve nesnenin kesin bir şekilde yerleştirilmesini hesaba katar. Modelden bağımsız yaklaşımlar, doğrudan kavrama adayları önerir ve tipik olarak yeni nesnelere genelleme yapmayı amaçlar. Ek yaklaşımlar tipik olarak ya kavrama adaylarını örnek alır ve onları bir sinir ağı kullanarak sıralar ya da doğrudan uygun kavrama pozları oluşturur [10–13]. Görsel tabanlı yaklaşımlar bir simülasyon ortamında mı, gerçek dünyada mı yoksa her ikisinde mi eğitildiklerine ve çeşitli sensör verilerini kullanmasına göre de farklılık gösterir. Ayrıca, yaklaşımlar ya açık ya da kapalı döngü şeklinde çalışmaya ilişkin de farklılık gösterir [5,14,15]. Görsel özelliklere dayalı sürekli geri bildirimin kullanılması, yaygın olarak görsel-servo olarak adlandırılır [15]. Bazı yaklaşımlar robot donanımının yanı sıra, kavrayıcı tipi ve kavrayıcı serbestliği göz önünde bulundurularak da farklılık göstermektedir. Bazı görsel sensör tabanlı yaklaşımlar yalnızca tek tek ayrılmış nesneleri kavramaya odaklanırken, diğerleri yoğun dağınıklık içinde kavramayı hedefler. Bazı yöntemler, nesneyi kavramak için daha iyi bir konfigürasyonda hareket ettirmekle kavrama öncesi manipülasyonları gerçekleştirebilir [3].

Tüm avantajlarına rağmen derin öğrenme, modellerinin eğitimi için büyük miktarda veri gerektirmesi dezavantajına sahiptir. Bu durum, özellikle gerçek sistemlerinde eğitim verilerinin üretilmesinin maliyetli ve zaman alıcı olduğu robotikte önemli bir sorun olarak karşımıza çıkmaktadır. Pinto ve arkadaşları bir robotu 700 saatten fazla bir sürede 50.000 deneme yaptırarak yeni nesneleri kavraması için eğitmiştir [16]. Levine, gürbüz bir kavrama performansı için 2 ayda

14 robot üzerinde paralelleştirilmiş 800.000 kavramadan faydalanmıştır [17]. QT-Opt çalışmasında, birkaç hafta içinde 7 robotta 560.000'den fazla kavrama örneği toplanmıştır [18]. Bu sistemlerde, kavrayıcı masa yüksekliğinin değiştirilmesi veya kameranın hareket ettirilmesi gibi donanım kurulumundaki değişikliklerden bağımsız değildir. Gürbüz robotik kavrama ve manipülasyon politikalarını öğrenmek için çok sayıda robotik kol kurma ihtiyacından kaçınmak için simülasyonları kullanmak cazip bir alternatif olarak karşımıza çıkmaktadır. V-REP/CoppeliaSim [19], PyRep [20], PyBullet [21], MuJoCo [22], Gazebo [23] ve Webots [24] yaygın olarak kullanılan fizik simülasyonlardır. Kullanım esnekliği ve yeterli miktarda veri kaynağı sağladıkları için simülasyonların kullanımı caziptir. Simülasyonların hızlı olması ve veri üretimi için birden fazla makinede paralel hale getirilebilmesi kullanımını cazip hale getiren diğer sebeplerdir. Simülasyonlar, robotların bileşenlerinde aşınma ve yıpranma olmadan ve ayrıca kullanıldıkları alanda herhangi bir kesintiye uğramadan eğitilmesini sağlar. Bu avantajların yanı sıra simülasyonlar, gerçek dünyanın özelliklerini tam olarak yakalayamaz.

1.3. Tezin Amacı

DPÖ yöntemleri, oyun alanında göstermiş oldukları başarılı sonuçlar sonrasında diğer alanlarda da kullanılmaya başlanmıştır. Bu tez çalışmasında, standart bir Derin Q Ağı (DQA) algoritmasının uygun hiperparametreler elde edilerek gerçekleştirilmesi ve robotik kavrama eylemi üzerinde geliştirilen yöntemin başarımı irdelenmiştir. Ayrıca birçok uygulamada karşılaşılan nesne kavrama problemlerine yönelik çözüm aşamalarının literatürde yer bulması amaçlanmıştır. Bu tezde güncel yaklaşımlardan biri olan pekiştirmeli öğrenmenin temel mantığının anlaşılması ve yapay zekânın gelişiminin önemli bir ölçütü olan robotikte uygulanması hedeflenmiştir. Bunun için insandaki kavrama tiplerinden en yaygın olan 6 kavrama tipini sınıflandırabilen bir derin öğrenme modeli geliştirilmiştir.

1.4. Tezin Yapısı

Bu tez çalışması 5 temel bölümden oluşmaktadır ve tezin yapısı aşağıdaki gibi düzenlenmiştir:

Bölüm 1 Giriş bölümüdür. Bu bölümde tezin amacı, kapsamı ve içeriği özetlenmiştir.

Bölüm 2'de genel hatlarıyla robotikten bahsedilmiştir. Robotiğin temelleri, karşılaşılan problemler ve uygulama alanlarının anlatımına yer verilmiştir.

Bölüm 3'te DPÖ konusu ele alınmıştır. Bu bağlamda, Yapay Sinir Ağları (YSA), Derin Öğrenme Yöntemleri ve DPÖ hakkında bilgiler verilmiştir.

Bölüm 4'te tez kapsamında yapılan uygulamalar sunulmuştur. İlk olarak yEMG sinyalleri kullanılarak kavrama sinyallerinin sınıflandırılması gerçekleştirilmiştir. Ardından DKQÖ (Derin

Konvolüsyonel Q Öğrenme) ile Pong Oyununu öğrenen bir uygulamanın nasıl gerçekleştirildiği anlatılmıştır. Son olarak düello DKQÖ algoritmasıyla simülasyon ortamında gerçekleştirilen robotik kavrama uygulamasının aşamalarına yer verilmiştir.

Bölüm 5 Sonuçlar bölümüdür. Bölüm 4'te gerçekleştirilen çalışmaların sonuç ve katkılarına detaylı olarak yer verildiği için bu bölümde genel bir değerlendirme yapılmıştır. Ayrıca, gelecek çalışmalara yönelik önerilerde bulunulmuştur.



2. ROBOTİĞE GENEL BİR BAKIŞ

Bu bölümde, nesne kavramının en önemli olduğu robotik teknolojisine ait temel bilgiler ve uygulama alanları ile ilgili bilgiler verilecektir. Ayrıca bu tez kapsamında ele alınan nesne kavrama problemi ile robotikte karşılaşılan diğer temel problemlere ait çeşitli bilgiler de bu bölümde sunulacaktır.

Robot kelimesi ilk defa 1920 yılında Karel Capek'in Rossum's Universal Robots adlı eserinde yer almıştır. Çekçede robota (işgücü) ve robotnik (işçi) gibi kelimelerden üretilmiştir [25]. Webster Sözlüğü'ne göre *görünüş ve davranış olarak bir insana benzemesi için yapılmış makinelere* denir. Yine Webster Sözlüğü'ne göre *bağımsız hareket edebilme (yürüme veya tekerlekler üzerinde yuvarlanma gibi) ve karmaşık eylemler (nesneleri kavrama ve hareket ettirme gibi) gerçekleştirme yeteneğine sahip canlılara benzeyen bir makinedir* [26]. Amerikan Robotik Enstitüsü'ne göre *çeşitli görevlerin yerine getirilmesi için değişken programlanmış hareketler yoluyla malzeme, parça, alet veya özel cihazları hareket ettirmek üzere tasarlanmış yeniden programlanabilir, çok işlevli bir manipülatördür* [27].

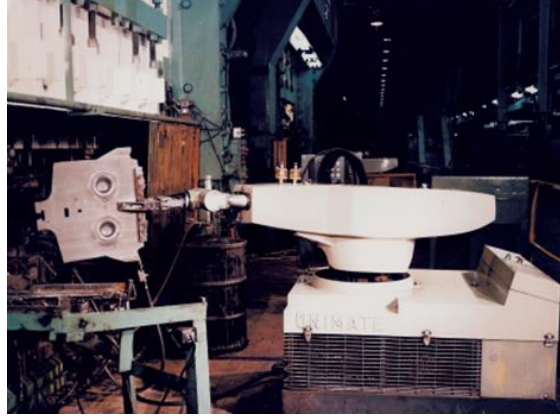
Robotik kelimesini ilk olarak bilim kurgu yazarı Isaac Asimov, akıllı robot davranışını araştırırken türetmiştir ve robotlar için günümüzde yaygın olarak bilinen üç yasasını açıklamıştır [28,29].

- 1) Bir robot, hiçbir şekilde insanoğluna zarar veremez veya pasif kalmak suretiyle zarar görmesine izin veremez.
- 2) Bir robot, kendisine insanlar tarafından verilen komutlara 1. yasa ile çelişmediği sürece itaat etmek zorundadır.
- 3) Bir robot, ilk iki yasayla çelişmediği sürece kendi varlığını korumak zorundadır.

20. yüzyılın ortaları, yapay zekâ alanında önemli gelişmelerin yaşandığı bir araştırma döneminin başlangıcı olup, insan zekâsı ve makineler arasındaki bağlantının ilk keşiflerini getirmiştir. Bu dönemde; ilk robotlar mekanik, kontrol, bilgisayar ve elektronik gibi farklı teknolojilerdeki gelişmelerden yararlanmışlardır. Her zaman olduğu gibi yeni tasarımlar yeni araştırmaları ve keşifleri teşvik etmiş, bu durum gelişmiş çözümlere ve dolayısıyla yeni kavramlara yol açmıştır. Zamanla bu anlamlı döngü, robot bilimi ve robot teknolojisi olarak adlandırılan robotik alanını doğuran bilgi ve anlayışı üretmiştir.

İlk robotik şirketi Unimation, 1956'da Devol ve Joseph Engelberger tarafından kurulmuştur ve Şekil 2.1'de görülen ilk endüstriyel robotu 1961'de geliştirmişlerdir [25]. Birinci nesil olarak isimlendirilen bu robotlar, üretim robotları olarak bilinen robotiğin bir alt sınıfıdır. Diğer alt sınıflar arasında temizlik, kişisel yardım veya tıbbi rehabilitasyon gibi hizmetler sağlayan hizmet robotları,

açık havada çalışan saha robotları ve bir insanın fiziksel formuna sahip insansı robotlar olarak birbirinden ayrılmaktadır.



Şekil 2.1. General Motors fabrikasında çalışan ilk Unimation robotu [25]

İlk robotlar, hassas üretim için sayısal kontrol makineleri ve uzaktan radyoaktif malzeme işleme için teleoperatörler olmak üzere iki teknolojinin bir araya gelmesiyle ortaya çıkmıştır. Bu efendi-köle (master-slave) kolları, insan kolunun mekaniğini bire bir kopyalamak için tasarlanmıştır ve ayrıca ilkel kontrole ve çevre hakkında çok az algıya sahip bir donanıma sahip olmuştur [15].

20. yüzyılın sonuna kadar entegre devrelerin, dijital bilgisayarların ve bileşenlerinin geliştirilmesi, bilgisayar kontrollü robotların tasarlanmasını ve programlanmasını sağlamıştır. Endüstriyel robotlar olarak adlandırılan bu robotlar, 1970'lerin sonlarında esnek üretim sistemlerinin otomasyonunda temel bileşenler haline gelmiştir. Otomotiv endüstrisindeki geniş uygulamalara ek olarak endüstriyel robotlar; metal ürünler, kimya, elektronik ve gıda gibi genel endüstrilerde başarıyla kullanılmıştır. Daha yakın zamanlarda, robotlar fabrikaların dışında temizlik, arama kurtarma, sualtı, uzay ve tıbbi uygulamalar gibi alanlarda yeni uygulamalarda kullanılmaya başlamıştır [6].

1980'lerde robotik, algı ve eylem arasındaki akıllı bağlantıyı inceleyen bilim olarak tanımlanmıştır. Bu tanımdan yola çıkarak bir robotik sistemin eylemi, ortamda hareket etmek için bir hareket aygıtıyla (tekerlek, bacak, pervaneler) veya çevrede bulunan nesneler üzerinde çalışmak için bir manipülasyon aygıtıyla (kol, uç işlevci, yapay el) uygulanmıştır. Algı; robotun durumu (konum ve hız) ve çevresindeki ortam (kuvvet ve dokunma, menzil ve görüş) hakkında bilgi sağlayan sensörlerden elde edilir. Akıllı bağlantı; robot ve ortamın algısına, mevcut modellerine dayanan ve öğrenme yeteneğinden yararlanan bir programlama, planlama ve kontrol mimarisine sahiptir [6].

1990'larda araştırmacıların tehlikeli ortamlarda insan güvenliğini sağlama, insan operatör yeteneğini geliştirme ve yaşam kalitesini artırma arzusuyla robotlara başvurma ihtiyacı desteklenmiştir. Bu tür uygulama senaryolarının ortak bir paydası, nihai olarak daha fazla yetenek

ve daha yüksek derecede özerklik gerektiren, nadiren yapılandırılmış bir ortamda çalışma ihtiyacı olmuştur [6].

Milenyumun başlangıcı itibariyle robotik, kapsam ve boyutlarda büyük bir dönüşüm geçirmiştir. Bu genişleme, alanın olgunluğu ve ilgili teknolojilerdeki gelişmelerle sağlanmıştır. Büyük ölçüde endüstriyel odaklı robotik, insan dünyasında karşılaşılan zorluklara çözümler sunarak hızla yayılmaktadır. Yeni nesil robotların evlerde, işyerlerinde ve topluluklarda insanlarla güvenli bir şekilde birlikte yaşaması, bununla birlikte hizmetler, eğlence, eğitim, sağlık, üretim ve yardım konularında destek sağlaması beklenmektedir [30].

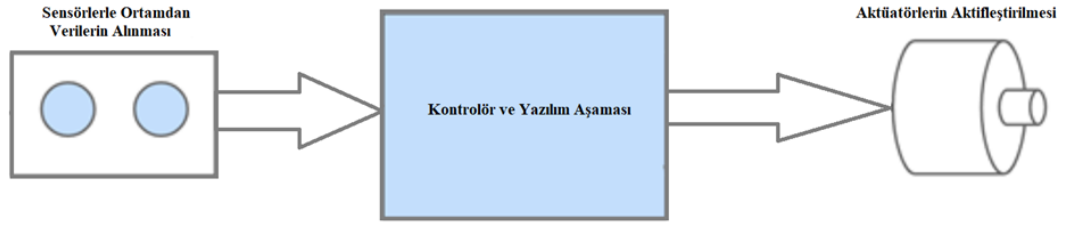
Genel olarak robot tanımı yapılacak olursa, önceden programlanmış bir görevi gerçekleştirmek için mantıksal eylemleri yerine getirebilen akıllı elektro-mekanik cihazlara denir. Robot, bir yazılım aracıyla sensörlerden gelen bilgilere dayanarak otonom kararlar veren ve aldığı bu kararları aktüatörlere aktaran bir mekanizmadan oluşmaktadır. Özü itibariyle, robotların tümü motor gibi çıkışlara, sensör gibi girişlere ve kodu işlemek veya çalıştırmak için bir kontrolöre ihtiyaç duymaktadır. Yani, basit bir robotta Şekil 2.2’de verildiği gibi aşağıdaki olaylar gerçekleşmektedir:

- Bulunduğu ortamın özelliklerini ölçmek ve örneklemek için girişlere (sensörlere) sahiptir.
- Ortamdaki durumunu değiştirecek motor, ışık, ses, valf, ısıtıcı gibi çıkışları vardır.
- Çıktılarını nasıl kontrol etmesi gerektiği konusunda otonom kararlar vermek için girdilerinden gelen verileri kullanan kontrolör vardır.

Robotiğin anlaşılması için temel konulardan olan robot kinematiği ve dinamiğinin iyi anlaşılması gerekmektedir. Ancak bu tez çalışmasında bu yöntemler yerine yapay zekâ yaklaşımları tercih edilmiştir. Sonraki alt başlıklarda bu konulara kısaca değinilecektir.

Robot kinematiği ve robot dinamiği, robotik sistemlerin nasıl modelleneceğine ilişkin genel bir bakış sağlamakta ve sistemleri kontrol etmek için bu modellerin nasıl kullanılacağı hakkında birtakım bilgiler vermektedir. Bu teknikler kullanılarak modellenen robot kolları, bacaklı ve tekerlekli makineler veya uçan sistemler gibi farklı robot türleri arasındaki benzerliklerin anlaşılmasını geliştirmeye çalışır. Çoğu robot, tek bir gövdeyle veya üzerinde farklı kuvvetlerin etki ettiği bir dizi gövdeyle tanımlanabilir. Ancak bu güçler farklı kaynaklardan gelebilmektedir. Serbest alanda hareket eden bir robot kolu, eklemlere etki eden aktüatör kuvvetleri tarafından tahrik edilir. Bacaklı bir robot ise ayrıca ayaklarında etkileşim kuvvetleriyle karşılaşır. Uçan araçlar da aerodinamik kuvvetler nedeniyle havada tutulur.

Robotik genel olarak, sabit tabanlı ve yüzer tabanlı sistemler olmak üzere iki kategoriye ayrılır. İlk kategori, zemine sağlam bir şekilde sabitlenmiş her türlü robotik makineyi içerir, buna örnek olarak endüstriyel robot kolları verilebilir. Bu tür sistemlerde çoğunlukla her eklemden bir aktüatör bulunur, böylece serbestlik derecesinin sistemlerdeki aktüatörlerin sayısına eşit olduğu anlamına gelir. İkinci kategori, mobil sistemleri yani hareketli bir tabanı olan robotları kapsar.



Şekil 2.2. Robotun basitleştirilmiş halinin şematik gösterimi

Robot kol örneğinin aksine, yüzer tabanın hareketi tipik olarak aktüatörler aracılığıyla doğrudan kontrol edilemez ancak yalnızca sistem üzerinde etkili olan temas veya aerodinamik kuvvetler gibi dış kuvvetler aracılığıyla kontrol edilebilir [15].

2.1. Robot Kinematiği

Kinematik, harekete neden olan kuvvetler/torklar dikkate alınmaksızın robotik bir mekanizmadaki cisimlerin hareketi ile ilgilidir. Robotik mekanizmalar özünde hareket için tasarlandığından; kinematik robot tasarımı, analizi ve kontrolü simülasyonunun en temel özelliğidir. Robotik topluluğu, temel kinematik problemlerini çözmek için farklı konum ve yönelim temsillerini ve bunların zamana göre türevlerini verimli bir şekilde uygulamaya odaklanır.

Robotik mekanizmalar, eklemlerle birbirine bağlanan katı cisimlerin sistemleridir. Katı bir cismin uzaydaki konumu ve yönelimi, topluca poz olarak adlandırılır. Bu nedenle robot kinematiği; bir mekanizmayı oluşturan cisimlerin pozunu, hızını, ivmesini ve pozunun tüm yüksek dereceli türevlerini tanımlar.

Robotikte gövde sistemlerinin bağlanabileceği birçok olası topoloji arasında, özellikle seri zincirler ve tamamen paralel mekanizmalar denilen iki topoloji önemlidir. Seri zincir, her biri yalnızca bir diğer üyeye bağlı olan ilk ve son üyeler hariç, her bir üyenin diğerine bağlı olduğu katı cisimler sistemidir. Tamamen paralel bir mekanizma, birden fazla eklemlerle birbirine bağlanan iki üyenin bulunduğu bir mekanizmadır. Uygulamada, her eklemin kendisi genellikle bir seri zincirdir [31].

2.2. Robot Dinamiği

Dinamik hareket denklemleri, robot mekanizmalarına etki eden çalıştırma ve temas kuvvetleri ile ortaya çıkan hızlanma ve hareket yörüngeleri arasındaki ilişkileri sağlar. Dinamik; mekanik tasarım, kontrol ve simülasyon için önemlidir. Bu uygulamalarda bir takım algoritmalar önemlidir ve ters dinamik, ileri dinamik, birleşik uzay atalet matrisi ve işlemsel uzay atalet matrisi gibi büyüklüklerin hesaplanmasını içerir [32]. Robot dinamiği, çalıştırma ve temas kuvvetleri arasındaki ilişkileri ve sonuçta ortaya çıkan hızlanma ve hareket yörüngelerini sağlar. Dinamik

hareket denklemleri; mekanik tasarım, kontrol ve simülasyonda faydalı olan bir dizi hesaplama algoritması için temel sağlar. Bu hesaplama algoritmalarının kullanımının büyüyen bir alanı, özellikle insan ve insansı modellerin kullanıldığı mobil sistemlerin bilgisayar animasyonlarıyla ilgilidir.

İleri beslemeli kontrolde ters dinamikler kullanılır. Simülasyon için ileri dinamikler gereklidir. Eklem alanı atalet matrisi, birçok ileri dinamik formülasyonunun ayrılmaz bir parçası olup analizde ve dinamikleri doğrusallaştırmak için geri besleme kontrolünde kullanılır. Operasyonel uzay atalet matrisi, görev veya uç işlevci seviyesinde kontrolde kullanılır.

2.3. Robot Mekanizması ve Tahriki

Robot mekanizmaları ve tahriki, robotik sistemlerin tasarımında ve yapımında izlenen temel prensipleri oluşturur [33]. Robotun kinematik denklemleri ve jakobyeni; hareket aralığını ve mekanik avantajını karakterize eder, boyutunun ve eklem düzeninin seçimini belirler. Bir robotun gerçekleştirmesi gereken görevler ve hareketinin ilgili kesinliği; mekanik yapı, aktarım ve aktüatör seçimi gibi ayrıntılı özellikleri belirler. Bir robotun hareketli iskeletini oluşturan kirişler, bağlantılar, dökümler, miller, kızaklar ve yataklar gibi fiziksel yapıya robotun mekanik yapısı veya mekanizması denir. Mekanizmanın bağlantılarının hareket etmesini sağlayan motorlara *aktüatör* denir .

İlk robotlar, çok çeşitli görevleri yerine getirebildikleri takdirde endüstrideki en büyük pazarı bulacakları varsayımıyla tasarlanmışlardır [15]. Ancak bu durumun hem maliyet hem de performans açısından pahalı olduğu görülmüştür. Günümüzde robotlar, artık belirli bir dizi görev düşünülerek tasarlanmaktadır. Robot tasarımında eklem sayısı, fiziksel boyut, taşıma kapasitesi ve uç işlevcinin hareket gereksinimleri göz önünde bulundurulur. Hareketli iskeletin konfigürasyonu ve robotun genel boyutu; erişim, çalışma alanı ve yeniden yönlendirme yeteneği için görev gereksinimlerine göre belirlenir. Bu özellikler, örneğin araç tasarım aşamalarından olan ark kaynağı ve boya püskürtmenin düzgün hareketi için gereken uç efektör yolu kontrolünün hassasiyetini etkiler.

Robotik sistem tasarımında önemli konulardan biri, robotun gerçekleştirmesi beklenen görevlerin aralığıdır. Robot, amaçlandığı çeşitli görevleri yerine getirmek için ihtiyaç duyduğu esnekliğe sahip olacak şekilde tasarlanmalıdır. Bu durum, robot mekanizmasının ve aktüatör sisteminin topolojisini belirler [33].

2.4. Robotik Algılama ve Tahmin

Algılama ve tahmin, herhangi bir robotik sistemin tasarımının temel unsurlarıdır. Robotikte temel olarak geri besleme kontrolü için robotun kendi durumu tahmin edilmelidir. Algı, sensör

verilerinin göreve yönelik yorumlanması olarak tanımlanır. Robotik bir sistemin kontrol edilmesinin basit olması için ortamın eksiksiz bir modelinin mevcut olması ve robot aktüatörlerinin bu modele göre hareket komutlarını mükemmel bir şekilde yerine getirebilmesi gerekir. Robotikte ilgilenilen birçok durumda eksiksiz bir dünya modeli mevcut değildir ve mekanik yapıların mükemmel kontrolü asla gerçekçi bir varsayım değildir. Algılama ve tahmin, bu eksikliği telafi etmenin bir yoludur. Algılama ve tahminin rolü, ortamın durumu ve robot sisteminin durumu hakkında kontrol, karar verme ve insanlar gibi ortamdaki diğer ajanlarla etkileşim için bilgi sağlamaktır.

Proprioepsiyon, robotun kendisinin durumunu kurtarmak için algılama ve tahmin olarak adlandırılır. Eksteroepsiyon, dış dünyanın durumunu kurtarmak için algılama ve tahmin olarak adlandırılır. Proprioepsiyon ve eksteroepsiyonun, birbirinden ayırt edilebilmesi gerekir. Uygulamada çoğu robot sistemi, kendi fiziksel durumlarını tahmin ve kontrol etmek için gerekli proprioepsiyona sahip olacak şekilde tasarlanmıştır. Öte yandan, sensör verilerinden ortamın durumunu kurtarmak genellikle çok daha büyük ve daha karmaşık bir problemdir. Sensör tabanlı robotların artık mobil gözetim, yüksek performanslı manipülasyon ve tıbbi müdahaleler gibi çeşitli uygulamalarda ortaya çıktığı göz önüne alındığında belirli bir sistem için uygun algılama ve tahmin; büyük ölçüde robota verilen göreve bağlıdır.

Algılama ve tahmin, fiziksel bir niceliği daha sonraki işlemler için kullanılabilecek bir bilgisayar gösterimine dönüştürme süreci olarak görülebilir. Bu nedenle algılama, bazı fiziksel varlıkları bir bilgisayar tarafından işlenebilen bir sinyale dönüştüren dönüştürücülere yakından bağlıdır. Algılama, aynı zamanda, ortamın görev odaklı bir modelinde duyuşsal bilgiyi temsil etme süreci olan algıyla da yakından bağlantılıdır. Bununla birlikte, sensör verileri genellikle bu işlemi karmaşıklaştıran çeşitli şekillerde bozulabilmektedir. Dönüştürücüden kaynaklanan istatistiksel gürültü, sayısallaştırma işleminde ayrıklaştırma yapılması ve zayıf sensör seçiciliği nedeniyle belirsizliğin ortaya çıkması şeklinde görülebilmektedir. Böylece, bilginin çevre modellerine uygun şekilde entegrasyonunu desteklemek ve sinyal-gürültü oranının iyileştirilmesi için tahmin yöntemleri ön plana çıkmaktadır [34].

2.5. Robotikte Hareket Planlaması

Temel robotik görevlerin birisi, karmaşık cisimler için bir statik engeller topluluğu arasında başlangıçtan hedef konumuna kadar çarpışmasız hareketler planlamaktır. Nispeten basit olmasına rağmen, bu geometrik yol planlama problemi hesaplama açısından zordur. Bu durum; otomatik planlayıcıların geliştirilmesini daha da karmaşık hale getiren belirsizlikler, geri besleme ve diferansiyel kısıtlamalar gibi gerçek robotların mekanik ve sensör sınırlamalarından geriye kalan ek sorunları hesaba katmaktadır. Modern algoritmalar, temel geometrik problemin zor örneklerini ele almakta oldukça başarılı olmuştur ve yeteneklerini daha zorlu örneklere de uygulamak için çok

uğraşmıştır. Bu algoritmalar; bilgisayar animasyonu, sanal prototip oluşturma ve hesaplamalı biyoloji gibi robotik dışındaki uygulamalarda yaygın bir başarıya sahiptir [35].

2.6. Robotikte Hareket Kontrolü

Robotikte hareket kontrolünü anlamak için basitçe bir robotun kontrolörünün, aktüatörlerden birine bir sonraki konuma doğru hızlanmasına neden olan bir sinyal gönderildiği kabul edilsin. Eklem, istenen hedefe ulaşır ulaşmaz hareketi ölçmek ve hareketi durdurmak için bir geri besleme sinyali kullanılsa bile istenen değeri aşabilir, ötesine geçebilir ve aktüatöre geri döndürmek için negatif bir sinyal gönderilmesini gerektirebilir. Pozisyon doğru bir şekilde elde edilene kadar bu ileri geri harekete devam edilir. Kararsız bir sistemde salınımlar küçülmez, aksine büyür ve sonunda arzulanan sistem kötü duruma gelebilir. Bunun nedeni, bağlantıların ve aktüatörlerin ataletle sahip olması ve sinyal kapatıldığında hemen durmamaları olabilir. Hedef noktaya yaklaşırken, hedef noktanın aşılması için aktüatörlere gönderilen gerilim ya da akım sinyalinin azaltılmasının mümkün olması gerekmektedir. Ancak bunun ne zaman ve ne oranda yapılması gerektiği ve sistemin kararsız hale gelmediğinden emin olunması gerekir. Ayrıca aktüatörlerin aşırıya kaçmadan istenildiği kadar hızlı hedefe ulaşması için ne oranda zorlanabildiğinin bilinmesi gerekmektedir. Tüm bunlar, robotun kontrol sisteminin arzulandığı gibi davranması için tasarlanmasıyla cevaplanan temel sorulardır.

Genel olarak, robot eklemlerindeki kuvvetlerin veya torkların doğrudan kontrolünün olduğu ve robotun dinamiklerinin bu kontrolleri eklem ivmelerine dönüştürdüğü varsayılmaktadır. Ancak bazı durumlarda, örneğin aktüatörlerin step motorlar olması halinde, eklem hızlarının doğrudan kontrolünün olduğu varsayılır. Böyle bir durumda bir eklemin hızı, doğrudan step motora gönderilen darbe dizisinin frekansı ile belirlenir. Hareket kontrol görevi, eklem uzayı veya görev uzayı olarak ifade edilebilir. Yörünge görev uzayında ifade edildiğinde, denetleyiciye sabit bir uç işlevci konfigürasyonları akışı verilir ve burada amaç, robotun bu yörüngeyi izlemesine neden olan eklem hızlarını komuta etmektir. Kontrolör, eklem uzayında istenen eklem konumlarının sabit akışıyla beslenir.

Step motor kontrollü robotlar, genellikle düşük veya öngörülebilir kuvvet/tork gereksinimleri olan uygulamalarla sınırlıdır. Ayrıca, robot kontrol mühendisleri, elektrik motorları için kullanıma hazır amplifikatörlerin hız kontrol modlarına güvenmezler, çünkü bu hız kontrol algoritmalarında, robotun dinamik bir modeli kullanılmamaktadır. Bunun yerine, robot kontrol mühendisleri, tork kontrol modunda amplifikatörleri kullanır. Amplifikatöre giriş, istenen tork veya kuvvet olarak alınır. Bu durum, robot kontrol mühendisinin kontrol yasasının tasarımında robotun dinamik bir modelini kullanmasına olanak tanır [36].

2.7. Robotikte Kuvvet Kontrolü

Bir manipölasyon görevinin başarısında temel unsur, bir robot ve ortamı arasındaki fiziksel teması idare etme yeteneğidir. Kaçınılmaz modelleme hatalarının ve belirsizliklerin temas kuvvetinin artmasına neden olması ve sonuçta etkileşim sırasında, özellikle katı ortamların varlığında kararsız bir davranışa yol açması sebebiyle saf hareket kontrolünün yetersiz olduğu görülmektedir. Güç geri beslemesi ve kuvvet kontrolü, bir robotik sistemin yetersiz yapılandırılmış ortamlarda sağlam ve çok yönlü bir davranışının yanı sıra insan varlığında güvenli bir şekilde çalışmasını sağlamak için zorunlu hale gelmektedir. Bir etkileşim görevinin belirlenmesi için uygun görev çerçevelerine göre görev geometrisi tarafından belirlenen doğal kısıtlamalar ve kontrol stratejisi tarafından belirlenen yapay kısıtlamalar belirlenir. Bu yaklaşım, hibrit kuvvet/hareket kontrol şemalarının sentezinin temel dayanak noktasıdır.

Telemanipölasyonla ilgili ilk çalışmalardan bu yana, insan operatöre bir yardımcı (slave) manipölatörü ile nesnelerin uzaktan işlenmesinde yardımcı olmak için kuvvet geri bildiriminin kullanımı tasarlanmıştır. Yakın zamanlarda, değış tokuş edilen kuvvetleri sınırlamak ve ortak olarak tutulan bir nesnenin sıkışmasını önlemek için iki veya daha fazla manipölatörün kontrol edildiğı işbirlikçi robot sistemleri geliştirilmiştir. Kuvvet kontrolü, açık uçlu ortamlarda robotik sistemlerin sağlam ve çok yönlü davranışının elde edilmesinde de temel bir rol oynar, öngörülemeyen durumlarda akıllı yanıt sağlar ve insan-robot etkileşimini geliştirir. Kuvvet, dokunma, mesafe ve görsel geri bildirim kullanan robotların, tipik endüstriyel atölye dışındaki yapılandırılmamış ortamlarda özerk olarak çalışması beklenir [37,38].

2.8. Robotikte Sistem Mimarileri ve Programlama

Robot yazılım sistemleri karmaşık bir yapıya sahiptir. Bu karmaşıklık, önemli belirsizlik ve gürültü karşısında çeşitli sensörleri ve aktüatörleri gerçek zamanlı olarak kontrol etme ihtiyacından kaynaklanmaktadır.

Robot sistemleri, beklenmedik durumları izlerken ve bunlara tepki koyarken verilen görevleri başarmayı amaçlamaktadır. Tüm bunları eşzamanlı veya eşzamansız olarak yapmak, sistem karmaşıklığına büyük ölçüde katkıda bulunur. İyi tasarlanmış bir mimarinin, mimariyi destekleyen programlama araçlarıyla birlikte kullanılması, genellikle bu karmaşıklığın yönetilmesine yardımcı olabilir. Tüm uygulamalar için en iyi olan tek bir mimari yoktur. Yani farklı mimarilerin farklı avantajları ve dezavantajları vardır. Belirli bir uygulama için mimari bir yaklaşım seçerken güçlü ve zayıf yönlerini anlamak önemlidir.

Robot mimarisi terimi genellikle birbiriyle ilişkili ancak mimari yapı ve mimari stil olmak üzere farklı iki kavramı ifade etmek için kullanılır. Mimari yapı, bir sistemin alt sistemlere nasıl bölündüğünü ve bu alt sistemlerin nasıl etkileşime girdiğini ifade eder. Bir robot sisteminin yapısı

genellikle kutular ve oklardan oluşan diyagramlar kullanılarak, gayri resmi olarak veya birleşik modelleme dili gibi teknikler kullanılarak daha resmi olarak temsil edilir. Buna karşılık, mimari stil, belirli bir sistemin altında yatan hesaplama kavramlarını ifade eder. Örneğin, bir robot sistemi yayımla-abone ol mesaj iletme tarzını kullanabilirken, diğeri daha senkronize bir istemci-sunucu yaklaşımı kullanabilir. Tüm robotik sistemlerde, belirli bir mimari yapı ve stil kullanılır. Ancak, mevcut birçok robot sisteminde kullanılan mimariyi tam olarak belirlemek zordur. Aslında, tek bir robot sistemi genellikle birkaç stili birlikte kullanır. Bunun nedeni kısmen sistem uygulamasının temiz alt sistem sınırlarına sahip olmaması ve mimari yapıyı bulanıklaştırmasıdır. Benzer şekilde mimari ve alana özgü uygulama genellikle birbirine sıkı sıkıya bağlıdır ve mimari tarzı bulanıklaştırır. Bu durum talihsiz bir durumdur çünkü iyi tasarlanmış ve anlaşılır bir mimari, robot sistemlerinin belirlenmesi, yürütülmesi ve onaylanmasında önemli avantajlara sahip olabilir. Genel olarak robot mimarileri, aşırı kısıtlayıcı olmaksızın robotik sistemlerin tasarımı ve uygulanması üzerinde faydalı kısıtlamalar sağlayarak geliştirmeyi kolaylaştırır. Örneğin; davranışları modüler birimlere ayırmak, anlaşılabilirliği ve yeniden kullanılabilirliği artırmaya yardımcı olur ve birim testini ve doğrulamasını kolaylaştırabilir [39].

2.9. Robotikte Yapay Zekâ Akıl Yürütme Yöntemleri

Çıkarım, planlama ve öğrenmeyi içeren yapay zekâ akıl yürütme teknolojisi, çok sayıda uygulamada başarılı bir şekilde kullanılmıştır. Otonom mobil robotlarda sembolik akıl yürütmenin gerekliliği ve yeterliliği hakkındaki tarihsel tartışmalara rağmen günümüzde bu tür robotların kontrol sistemindeki bir kısmın akıl yürütmeyi içermesi gerektiği veya içerebileceği konusunda bir anlaşma var gibi görünmektedir. Hibrit kontrol mimarileri, robotun dinamik bir ortamda güvenli bir şekilde hareket edebilmesi için yeterince kısa kontrol döngüsü sürelerini garanti ederken katkılarını kontrolörün diğer parçalarıyla birleştirme gibi zor bir problemi çözmek için tipik yazılım yapıları olacaktır.

2.9.1. Bilgi Temsili ve Çıkarım

Akıl yürütme, akıl yürütenin/robotun üzerinde akıl yürüteceği ortamının bölümlerinin veya yönlerinin açık bir temsiline sahip olmasını gerektirir. Burada iki soru ile karşılaşılır. Birinci soru, böylesine açık bir temsil için uygun formatlar nelerdir ve temsil edilen bilgi nereden gelebilir? İkinci soru, robotun ortamının veya en azından bir kısmının, önceki bir sembolik açıklamaya ve sensörlerden ve diğer ajanlarla iletişimden elde edilen en son ortam bilgisine dayalı olarak gerçek zamanlı olarak sembolik bir tanımını üretme ve sürdürme problemine atıfta bulunur. Bir robottaki pratik sembolik akıl yürütme, bilginin yeterince güncel tutulabilen kısmıyla sınırlıdır. Bu durum, açık bir şekilde ortam hakkında statik bilgiyi içerir. Kamera verilerinden nesne tanıma önemli bir

yöntemdir. Bilgi temsili ve çıkarım, bilgiyi temsil etmeye uygun formüller gibi birinci soruya verilen yanıtlarla ilgilenir. Bilgi temsili, belirli bir alan hakkında bilgiyi ifade etmek için hem epistemolojik hem de sayısal olarak yeterli olan formülasyonların tasarımına odaklanan yapay zekâ alanıdır.

Akıl yürütme yöntemleri, bir dereceye kadar mantığın veya bayesçi olasılık teorisinin temel formülasyonlarının çeşitlemelerini kullanır. Her iki formülasyon, özellikle de varyantlar dahil edildiğinde herhangi bir bilgi temsili uygulamasına katkıda bulunabilirler. Bununla birlikte, bir robot kontrol sistemindeki akıl yürütme, diğer bilgi tabanlı sistemlerdeki akıl yürütmeden farklıdır. Bilgiye dayalı bir robot, tipik olarak oldukça sınırlı kaynaklara sahip dinamik bir ortamda kapalı döngü kontrolünde çalışan gömülü bir sistemdir. Yani kendi sensör girdisini kendisi yorumlamalı ve uygun gördüğü eylemlerin en azından bazılarını bağımsız olarak gerçekleştirmelidir. Robotlarda temsil edilmeyen bazı zekâ biçimlerinin elde edilebileceği ikna edici bir şekilde kanıtlanmıştır. Semboller açısından akıl yürütme, gerçek sensör verileri ve aktüatör kontrolünde temellendirilemiyorsa en karmaşık temsil ve çıkarım sisteminin bir robot için hiçbir faydası yoktur. Öte yandan temsil ve çıkarım; saha, hizmet ve müdahale gibi uygulamalarda robotların performansını, yazılım tasarımını ve kullanıcı arayüzünü iyileştirir [40,41].

2.9.2. Robotikte Eylem Planlama

Akıl yürütme yöntemleri, eylem planlama ve öğrenme olmak üzere iki genel uygulamaya ayrılabilir. Geleneksel yapay zekâ anlamında planlama, bir ajanın belirli bir hedefler kümesine ulaşması için yapması gereken bir eylem planı hakkında müzakere etme anlamına gelir. Programlama (scheduling), zaman ve potansiyel olarak diğer kaynakları, belirli son teslim tarihlerine ve kaynak kısıtlamalarına uyulacak şekilde bir dizi eyleme tahsis etmek anlamına gelir. İki faaliyet kavramsal olarak farklıdır ve tarihsel olarak genellikle önce planla, sonra programla şeklinde kademeli olarak ele alınmıştır. Son zamanlarda, daha verimli zamansal akıl yürütme yöntemlerinin ve daha verimli planlama algoritmalarının ortaya çıkmasıyla, bunlara yönelik bütünleşik bir şekilde ilgilenme eğilimi vardır. Bunun mantığı, önce bazı ölçülere göre optimal bir plan oluşturmanın ve daha sonra bu plan için optimal bir programlama tasarlamının zaman ve kaynak açısından optimal bir programa yol açmaz. Bununla birlikte kombinasyonları her zaman gerekli olmadığından eylem planlama ve öğrenmeyi ayrı ayrı düşünmek yine de mantıklıdır [42].

Yapay zekâdaki robot planlaması veya plana dayalı robot kontrolü terimi, tipik olarak otonom bir mobil robotun kontrol yazılımında, planlama yöntemlerinin kullanımını ifade eder. Robotikteki mevcut hareket veya yol planlama yöntemlerinin yerini alması değil, daha genel eylem planlaması için bunları tamamlaması amaçlanmıştır. Planlamayı robot kontrolünde kullanmanın birkaç amacı vardır, bunlar robot tarafından otomatik olarak çevrimiçi oluşturulmuş olabilir. Eylem planlamasının yardımcı olabileceği amaçların ve soyutlama düzeylerinin çokluğuna karşılık olarak,

robot planlaması için tek bir yöntem, biçimcilik veya yaklaşım yoktur. Bunun yerine planlama yöntemlerinden herhangi biri kullanılabilir. Sonuç olarak robot planı terimi; robotun programını, robotun gelecekteki yürümesinin açıkça nasıl yapıldığının belirtir. Planlama metodolojisi açısından robot planlamasının özgünlüğü, robotun benimsediği plan ne olursa olsun kapalı döngü yürütme ve izleme gereksiniminden gelir. Teknik olarak bu durum, robot planlamasında, eylem biçimciliğinde ve robot kontrol mimarisinde kısıtlamalar getirir. Robot kontrol mimarisi tasarımında, planlama ve planların genel robot kontrolüne uygun şekilde entegre edildiğinden emin olunmalıdır [42,43].

2.9.3. Robotlarda Öğrenme

Yapay zekânın önemli alanlarından biri olan makine öğrenimi ile öğrenme ve zekâ arasındaki bağlantı açıkça görülmektedir. Öğrenme programının ilk örneklerinden biri, 1959'da Samuel'in Dama programıydı [41]. Öğrenmenin tanımı, zekânın tanımı kadar geneldir. Temel olarak öğrenme; sistemin kendi performansını veya bilgi birikimini tecrübeye dayalı olarak iyileştirme yeteneği olarak ifade edilebilir. Yapay zekâda ana klasik öğrenme şemaları; tümdengelim, tümevarım veya analogi yoluyla yeni bilgi üretmek için sembolik temsilleri manipüle etmeye dayanır. İstatistiksel öğrenme, örneğin robotik ve diğer alanlarda kullanılan yaygın bir öğrenme şeması olan örneklerden modeller çıkarmak için verilerin sınıflandırılmasını sağlayan yöntemleri kapsar. Gösterimler olasılıklı olduğunda, Bayesian akıl yürütme kullanılır. Biyolojik nöronların resmi modellerine dayanan sinir ağları, istatistiksel öğrenme ile ilgili veri sınıflandırma tekniklerini kapsar. Robotikte bir başka popüler öğrenme şeması, Markov Karar Süreçleri (MKS) ve dinamik programlama ile yakından ilişkili olan pekiştirmeli öğrenmedir. Esas olarak, doğru bir yanıtı güçlendirmek için kullanılan her robot eylemiyle ilişkili bir pekiştirme sinyaliyle dayanır. Öğrenme, denetimli veya denetimsiz olarak da sınıflandırılabilir [44]. Bu durum, sistemin hareketinden sonra sistemi ödüllendirmek için açıkça doğru cevabı verebilecek bir denetmenin olup olmamasına veya bunun için sadece sistemin kendi performansının mevcut olup olmamasına bağlıdır. Ana denetimli sembolik öğrenme şemalarından biri mantık programlamadır. Bu süreç, mantık temsillerine ve birinci dereceden mantık çıkarımına dayanır ve bağlamsal bilgiyi kullanır.

Tümevarımsal mantık programlamanın amacı; doğru veya yanlış değerler döndürdükleri ve arka plan bilgisi olarak adlandırılan bir dizi değişken verilen mantık programlarını sentezlemektir veya öğrenmektir. Buradaki zorluk, verilerle uyumlu minimal (gerekli ve yeterli) bir program bulmaktır. Pozitif değişkenler için doğru, negatif olanlar için yanlış döndürülmektedir. Bu program aynı zamanda genelleme yapabilmelidir, yani yeni örneklerle uygulandığında doğru cevabı üretebilmelidir. Genel olarak, tümevarımsal mantık programlama izlenebilir bir teknik değildir [45].

İstatistiksel öğrenme; robotik ve bilgisayarla görü alanında çok yaygın olarak kullanılmaktadır [46]. Değerleri modelin bir örneğini sağlayan rastgele değişkenler olarak kabul

edilen verilerden modelleri öğrenmek amacıyla modelleri kategoriler halinde sınıflandıran Bayesian öğrenme, çekirdek yöntemleri ve sinir ağları gibi çeşitli teknikleri kapsar. İlk olarak, algoritmanın bir dizi veri ve ilişkili etiketlerle (denetimli öğrenme) sunulduğu bir eğitim dönemi bulunmaktadır. Ardından algoritma verileri kendi başına etiketler. Eğitim aşamasının amacı, sınıflandırma hatasını en aza indirmektir.

Bayesian öğrenme; veri ile beslenen hipotezlerin olasılığını hesaplamak için Bayes kuralının uygulanmasıdır [47]. Eğitim süresi, her sınıf için verilerin olasılığını sağlar. Sınıflandırma ayrıca Bayes kuralını uygulamak için önceki sınıf dağılımının bilinmesini gerektirir. Maksimum olabilirlik ve beklenti maksimizasyonu dahil olmak üzere Bayes öğrenmenin çeşitli varyasyonları bulunmaktadır. Bayes teknikleri orta hesaplama maliyetine sahiptir ve uygulanması kolaydır, ancak çevrimiçi değildirler.

Sinir ağları; ağırlıklı bağlantılarla birbirine bağlanan temel hesaplama birimlerinden (nöronlardan) oluşur [48]. Her birim, girdilerinin ağırlıklı toplamını hesaplar ve girdi belirli bir eşikten büyükse etkinleştirilir. Dolayısıyla bu işlem, bir sınıflandırıcının temelini sağlar. En yaygın olarak kullanılan ileri beslemeli bir ağın temel yapısı, bir girdi katmanı, bir veya daha fazla gizli katman ve bir çıktı katmanı içerir. Bağlantı ağırlıkları aracılığıyla bir sinir ağının çıktısı, girdilerinin bir fonksiyonudur. Sinir ağının eğitimi, verileri doğru bir şekilde sınıflandırmak için ağırlığın ayarlanmasından oluşur. Bayesçi öğrenmenin aksine sinir ağları sınıflandırıcı hakkında hiçbir fikir vermez, ancak çevrimiçi olarak kullanılabilirler.

Pekiştirmeli öğrenme; ajanın bir ödül sinyali veren ortamda hangi eylemleri yapması gerektiğine karar verebilen bir makine öğrenmesi türüdür. Pekiştirmeli öğrenmede robotun veya ajanın amacı, en iyi politikayı yani belirli bir hedefe ulaşmak için eylemlerin sırasını öğrenmektir. Bunu yapmak için kümülatif ödülü maksimize etmesi gerekir. Bu nedenle pekiştirmeli öğrenmenin genel çerçevesi markov karar süreçleriyle aynıdır. Robot, ortam durumlarını algılar ve bir politika oluşturan sıralı eylemleri gerçekleştirip sonrasında bir durumdan diğerine geçiş olasılıkları vardır. Robot, zaman içinde beklenen kümülatif ödülleri ifade eden bir değer fonksiyonunu maksimize etmeye çalışır. Öğrenme süreci göz önüne alındığında, sürekli eylem uzaylarına uygulanması kolay değildir. Pekiştirmeli öğrenmeyi sürekli uzaylara uygulamak için genellikle örneklemeye başvurulur [49,50].

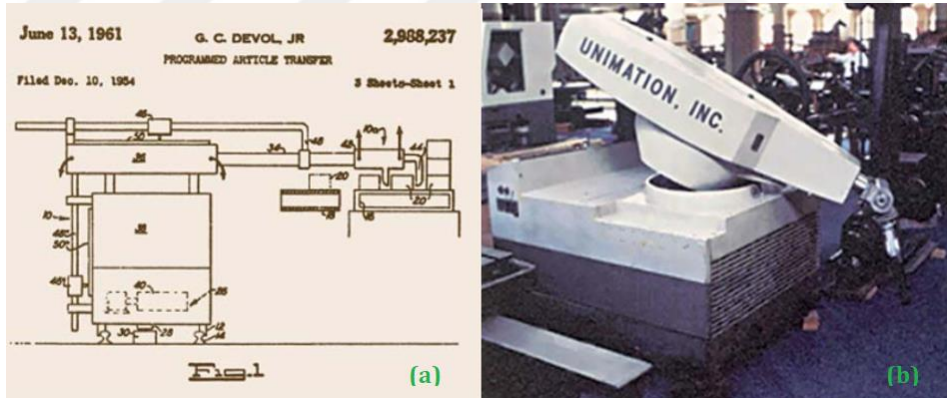
2.10. Robotik Uygulama Alanları

Robotik uygulama alanları, endüstriyel robotlardan çeşitli hava, kara, deniz ve uzay uygulamalarına ve eğitim robotiğine kadar değişen ve büyüyen bir teknoloji alanını kapsamaktadır [51]. Robotik için gelecek vizyonuna bakıldığında, robotların her alanda yaygın bir şekilde kullanılması amaçlanmaktadır. Geleceğin robotlarının; tüm tehlikeli, kirli ve kasvetli görevleri

yerine getireceği öngörülmektedir. Günümüzde teknolojinin ulaştığı düzeye bakıldığında, robotlar artık fabrikalardan saha ve hizmet uygulamalarına doğru ilerlemeye başlamıştır.

2.10.1. Endüstriyel Robotik

Günümüzdeki çoğu robotun temeli ilk endüstriyel robot tasarımlarına dayanmaktadır. Robotları daha insan dostu ve farklı uygulamalara uyarlanabilir yapan teknolojinin çoğu, endüstriyel robot üreticilerinden çıkmıştır. Endüstriyel robotlar, günümüzde robot teknolojisinin açık ara en büyük ticari uygulamasıdır. Robot kontrolünün tüm önemli temelleri, başlangıçta endüstriyel uygulamalar düşünülerek geliştirilmiştir. Endüstriyel robotun icadı, George Devol'un Şekil 2.3'te görülen *programlanmış bir ürün transferi* için patent başvurusu yaptığı 1954 yılına uzanmaktadır. İlk robot şirketi olan Unimation, George Devol ile Joseph Engelberger birlikte çalışmaya başladıktan sonra kurulmuştur. Unimation şirketi ilk robotu 1961'de bir döküm makinesinden parça çıkarmak için Şekil 2.3 General Motors fabrikasında hizmete sunmuştur [15,52]. Şekil 2.3 (a)'da, G. Devol ve J. Engelberger'in evrensel ve otomasyon kelimelerinin İngilizce bir birleşimi olan ilk robot şirketi Unimation'ı oluşturmak için ortak çabalarının başlangıcı olan patent görülmektedir. Şekil 2.3 (b)'de ise robotun 1961'de General Motors fabrikasında oldukça basit bir taşıma görevini gerçekleştirdiği görülmektedir. Sonrasında diğer birçok şirket endüstriyel robotlar geliştirmeye ve üretmeye başlamıştır.



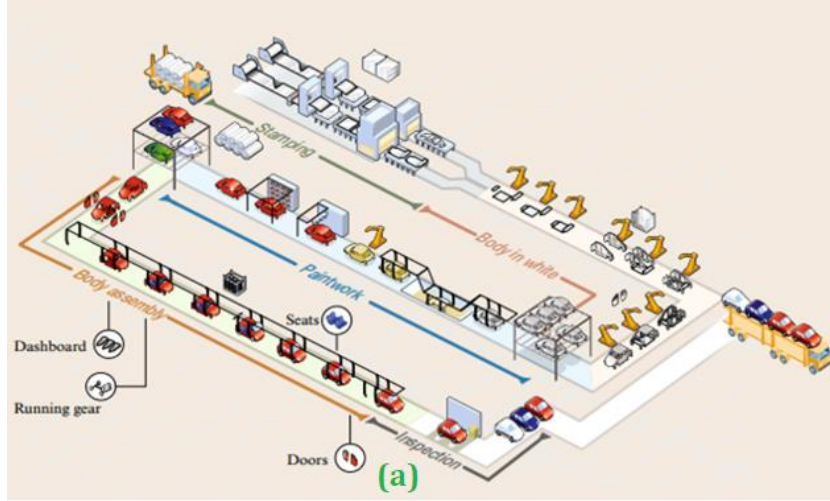
Şekil 2.3. Unimation şirketi tarafından geliştirilen ilk endüstriyel robotun (a) patenti ve (b) kendisi[15]

Endüstriyel robotlar, yüksek üretkenliği, kaliteyi ve uyarlanabilirliği minimum maliyetle oluşturulması amaçlanmıştır. Endüstriler ve ortaya çıkan üretim süreçleri, giderek artan bir şekilde gelişmiş robot teknolojisine olan bağlılığı arttırmıştır. Bu endüstrilerin robot kurulumlarının sayısındaki payı istikrarlı bir şekilde artmaktadır. Bir yandan endüstriyel robotların üretimi ve diğer yandan robot çalışma hücrelerinin planlanması, entegrasyonu ve işletilmesi büyük ölçüde bağımsız mühendislik görevleridir. Yeterince büyük miktarlarda üretilebilmesi için, bir robot tasarımının en geniş potansiyel uygulama setinin gereksinimlerini karşılaması gerekir. Pratikte bunu başarmak zor olduğu için, montaj, paletleme, boyama, kaynak, işleme ve genel taşıma görevleri gibi uygulama

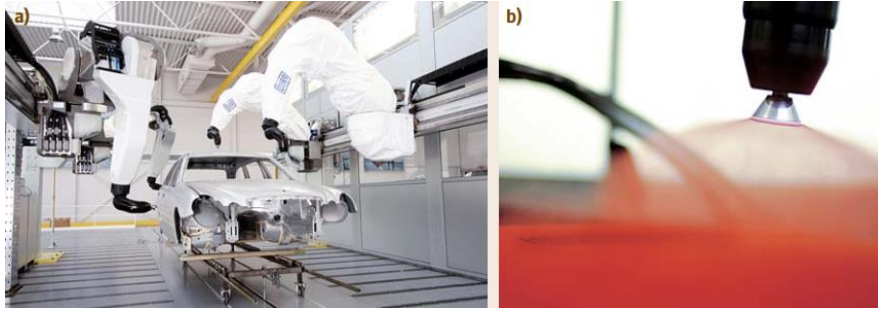
kategorileri için taşıma kapasitesi, robot eksen sayısı ve çalışma alanı hacmi ile ilgili çeşitli robot tasarım sınıfları ortaya çıkmıştır [53]. Bir robot çalışma hücresi, genellikle önemli mühendislik uzmanlığı gerektiren özelleştirilmiş planlama, entegrasyon, programlama ve konfigürasyonun sonucudur. Standartlaştırılmış mühendislik yöntemleri, araçları ve en iyi uygulama örnekleri, maliyetleri azaltmak ve daha öngörülebilir performans sağlamak için kullanılabilir hale gelmiştir.

Günümüzün endüstriyel robotları, esas olarak otomotiv, elektronik ve elektrikli eşya endüstrileri tarafından tanımlanan sermaye yoğunluklu büyük hacimli üretim gereksinimlerinin bir sonucudur. Geleceğin endüstriyel robotlarında, özellikler ve performans verileri açısından günümüz tasarımlarının yalnızca bir tahmininden ibaret olmaktan ziyade daha geniş bir uygulama alanı ve endüstri yelpazesine hitap eden yeni tasarım ilkeleri takip edilecektir. Aynı zamanda, özellikle bilgi teknolojisi ve yapay zekâ dünyasından gelen yeni teknolojilerin, gelecekteki endüstriyel robotların tasarımı, performansı ve maliyeti üzerinde artan bir etkisi olacaktır. Endüstriyel robot uygulamaları, endüstriyel kaynak, endüstriyel araç montajı, endüstriyel boyama, malzeme aktarım otomasyonu, makineyle işleme ve görevlerin yürütülmesinde insan-robot işbirliği olmak üzere farklı başlıklarda incelenir [54,55]. Robot otomasyonunda; endüstriyel araç montajı, ilk zamanlardan beri öne çıkan robot uygulamalarından biri haline gelmiştir [56]. Şekil 2.4'te endüstriyel araç montajı süreci görülmektedir. Şekil 2.4 (a)'da, bir araba gövdesi montajı adımları görülmektedir. Şekil 2.4 (b)'de, Almanya Bremen'deki Mercedes C sınıfı aracın üretim hattı ve transfer hattında gövde boyunca montaj robotları görülmektedir. Araba gövdelerini taşıyan düzenekler robot bahçesinden geçmektedir.

İnsan operatörler için tehlikeli çalışma koşulları, bir Norveç şirketi olan Trallfa'yı 1969'da özellikle otomotiv endüstrisindeki tamponları ve diğer plastik parçaları püskürtmek için basit spreyci boyama robotları geliştirmeye teşvik etmiştir [57]. Başlangıçta spreyci boyama robotları, insan işçilerden kopyalanan hareketleri yerine getirirken, son teknoloji programlama sistemleri boya birikimini, kalınlığını ve kapsamını optimize etmek için entegre süreç simülasyonları sunduğundan, günümüzde endüstriyel robot boyama için programlamanın çoğu çevrimdışı olarak yapılmaktadır. Şekil 2.5'te bazı boyama robotları görülmektedir.



Şekil 2.4. (a) Araba gövdesi montajı adımları ve (b) üretim hattında görev yapan montaj robotları



Şekil 2.5. Boyama robotları. (a) Araba gövdesi boyama için çok robotlu bir çalışma hücresi (b) Yüksek hızlı dönen boya tabancası

2.10.2. Sualtı Robotiği

Deniz taşıtları, uzun zamandır insanlar tarafından kullanıldığı bilinmektedir ancak su altına girebilen araçların çok eski bir tarihi olmayıp yeni sayılmaktadır. Leonardo Da Vinci, bir sualtı aracı tasarlayan ilk kişi olabilir. Bu yöndeki girişimleri, 1480 ile 1518 yılları arasında yazılan Codice Atlantico'da kaydedilmiştir [57].

Sualtı robotları günümüzde bir dizi bilimsel, ticari ve askeri görevde önemli roller oynamaktadır [58,59]. Uzaktan kumandalı araçlar, tüm bu alanlarda çok iyi bir şekilde kullanılmaya başlamıştır ve insan operatörlerin üzerindeki yükü hafifletmek ve performansı artırmak için giderek daha fazla otomatik hale gelmektedir. Otonom sualtı araçları neredeyse yalnızca araştırma çalışmaları için kullanılmaktadır ancak numune alma ve diğer müdahale görevleri daha uygulanabilir hale gelmiştir [60].

Askeri çalışmalar, sualtı robotik yeteneklerinin geliştirilmesinde her zaman lider olmuştur. Ayrıca test silahlarının kurtarılması ve derin deniz kurtarma gibi görevler için uzaktan kumandalı araçların geliştirilmesine öncülük etmişlerdir. Günümüzün ticari ve bilimsel uzaktan kumandalı araçları, doğrudan bu sistemlerden türetilmiştir [61,62]. Birçok farklı ülke, otonom sualtı araçlarını askeri araştırmalar, çevresel veriler toplama ve mayın gibi tehlikeleri araştırmak için kullanmaktadır. Geliştirilmekte olan otonom sualtı araçlarının yalnızca mayınları tespit etmekle kalmayıp, onları devre dışı bırakabilmesi de amaçlanmaktadır.

Daha yenilikçi konseptler de geliştirilme aşamasındadır. Bunlar, geleneksel su üstü gemileri ve denizaltıların uzantıları olarak hareket edebilen ve su üstü gemileri, denizaltılar ve uçaklarla elde edilebilecek maliyetlerden çok daha düşük maliyetlerle geniş alanlar üzerinde uzun süre boyunca gözetim sağlayan otonom sualtı araçları ağırlarını içermektedir.

2.10.3. Hava Robotiği

Hava robotiği terimi, akıllı ve küçük boyutlu uçan makinelerin yeni bir sınıfını temsilen Robert Michelson tarafından kullanılmıştır [63]. Bununla birlikte, hava robotiği altında yer alan sistem ve faaliyetlerin kapsamının çok daha fazla genişleyebileceği ve başlangıcı havacılığın doğuşuyla birlikte 20. yüzyılın başlarına dayanmaktadır.

Uçuşun sunduğu hareketlilik düzeyine sahip robotlar için çok çeşitli potansiyel uygulamalar mevcuttur. Hava robotiğinin askeri uygulamaları, motorlu uçuşun başlangıcından bu yana kabul görmüştür ve gözetleme, hedefleme ve hatta saldırı görevlerinde bazen olağanüstü bir etkiye sahip oldukları fark edilmiştir. Sivil uygulamaların kapsamı daha da geniştir ve uzaktan algılama, afet müdahalesi, görüntü elde etme, gözetleme, nakliye ve ürün teslimatını içermektedir [64,65].

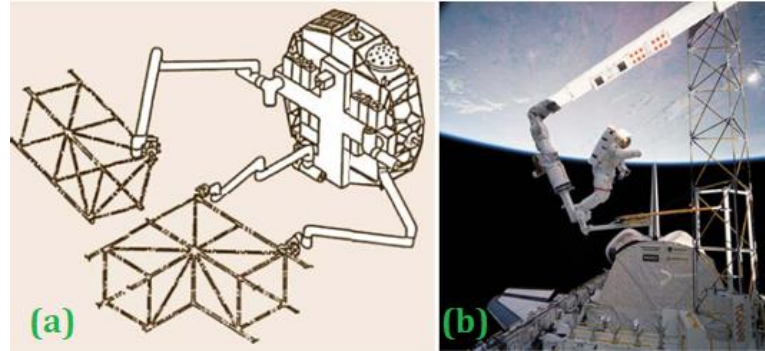
Hava robotlarının karşılaştığı zorluklar, insan-makine arayüzü tasarımı sorunları, navigasyon, güvenlik/güvenilirlik, çarpışma önleme ve kalkış/iniş teknikleri gibi çeşitli ve farklı alanları kapsamaktadır. Hava robotlarının boyutu, uçuş dinamiklerini önemli ölçüde etkilemekte ve küçük hava robotları daha büyük muadillerinden daha farklı görünebilmektedir. İnsanlı muadillerine benzer şekilde, hava robotları çeşitli tahrik sistemlerinden faydalanabilmekte ve geniş hız aralıklarında çalışabilmektedir. Hava robotlarının kontrollü uçuş yapabilmeleri için güvenilir

pozisyon ve harekete geirme ekipmanı ile donatılması gerekmektedir ve bu durum, bu alanda arařtırma veya geliřtirme yapmadan nce nemli bir gereklilik teřkil etmektedir [66].

2.10.4. Uzak Robotları Ve Sistemleri

Uzak topluluęunda, herhangi bir insansız uzak aracı robotik uzak aracı denir. Bununla birlikte uzak robotları, astronotların asistanı olarak yrngede maniplasyon, montaj veya servis iřlevlerini kolaylařtırabilen veya insan kařıflar iin yardımcı olarak uzak gezegenlerdeki keřif alanlarını ve yeteneklerini geniřletebilen daha yetenekli cihazlar olarak kabul edilmektedir. Yrnge ortamında kullanılan ilk robotik maniplatr kolu, uzak mekięi uzaktan maniplatr sistemi olmuřtur. 1981 yılında STS-2 grevinde bařarıyla gsterilmiřtir ve bugn hala alıřır durumdadır. Bu bařarı yrnge robotięinde yeni bir aę amıřtır ve arařtırmacılara yeni konseptlerin geliřtirilmesinde ilham olmuřtur. Bir telerobotik hizmet saęlayıcının kavramsal tasarımı řekilde grlmektedir. Mekik kargo blmesinde bir astronotun ara dıř aktivitesi iin bir platform olarak kullanılan uzak mekięi uzaktan maniplatr sistemi řekil 2.6'da grldę gibidir. Uzak robotları ve sistemlerindeki uygulamalar temel olarak ařaęıdaki gibi karakterize edilebilir [70–73].

Maniplasyon: Robotikte temel bir teknoloji olmasına raęmen, yrnge ortamındaki mikro yerekimi, maniplatr kollarının ve iřlenen nesnelerin hareket dinamiklerinde nemlidir. Ana gvdeyi etkileyen tepki dinamikleri, robotik elin kullanılacak bir nesneye temas etmesi durumundaki darbe dinamikleri ve yapısal esneklikten kaynaklanan titreřim dinamikleri maniplasyon konuları ierisinde yer almaktadır.



řekil 2.6. (a) Bir telerobotik hizmet saęlayıcının kavramsal tasarımı [67], (b) uzak mekięi uzaktan maniplatr sistemi [68]

Hareketlilik: Uzak bir gezegenin yzeyinde seyahat eden keřif robotlarında hareket kabiliyeti nemlidir. Bu yzeyler doęal ve przldr ve bu nedenle geiři zordur. Algılama (sensing) ve algı (perception), ekiř mekanięi ve ara dinamięi, kontrol ve navigasyon doęal bir ortamda olması gereken mobil robotik teknolojileridir.

Teleoperasyon ve zerklik: Uzaydaki bir robotik sistem ile dnyadaki kontrol odasında bulunan insan operatr arasında nemli bir zaman gecikmesi olabilmektedir. Bu nedenle

telerobotik teknolojisi, uzay robotiğinde vazgeçilmez bir bileşendir ve otonom yöntemlerinin çözüm olarak getirilmesi makul bir sonuçtur.

Ekstrem ortamlar: Manipülatör dinamiklerini etkileyen mikro yerçekimi ortamına veya yüzey hareketliliğini etkileyen doğal ve engebeli araziye ek olarak, zorlu ve pratik mühendislik uygulamalarına olanak sağlamak için çözülmesi gereken aşırı uzay ortamlarıyla ilgili bir dizi sorun vardır. Bu tür sorunlar arasında aşırı yüksek veya düşük sıcaklıklar, yüksek vakum veya yüksek basınç, aşındırıcı atmosferler, iyonlaştırıcı radyasyon ve çok ince toz bulunmaktadır.

2.10.5. Tarım ve Ormancılıkta Robotik

Robotik, tarım ve ormancılıkta önemli bir etki yaratmıştır [69]. Çiftçiler, topraklarına verilen zararı en aza indirmek için otomatik araç rehberliğine ihtiyaç duymaktadırlar. Hayvancılık prosedürlerinde önemli enstrümantasyon ve mekanizasyon vardı. Günümüzde ürünün otomatik olarak algılanması, taşınması ve işlenmesi artık yaygınlaşmıştır. Ormancılıkta, eşzamanlı konum belirleme ve haritalama teknikleriyle budama ve yönlendirme otomasyonu endüstride çözüm bekleyen konular arasındadır. Yapay görme ile uydu aracılığıyla küresel konumlandırmanın birleşimi, bir traktörün sıranın sonunda sürülmemiş arazi dönüşünü gerçekleştirerek bir sıra mahsulü takip etmesine olanak tanır. Ürünlerin renk sınıflandırması ve derecelendirilmesi tek başına bir algılama problemi değildir ancak onu robotiğin çalışma alanına yerleştiren bir ürün işleme ölçüsü içerir. Robotik bu alanlara hızlı bir giriş yapsa da, bazı uygulamalarda yeni teorik tekniklerin geliştirilmesini gerektirecek kadar yeterli verimlilik elde edilmemiştir. Akıllı robotiğin kullanımı bu alanda etkili olacaktır [70].

2.10.6. İnşaat Robotik

Yüzyıllar boyunca, üretim verimliliğini artırmak için inşaat mühendisliği alanına ve inşaat endüstrisine çeşitli makine sistemleri entegre edilmiştir [71]. Tarım, madencilik ve ormancılık alanlarıyla ortak olarak, uzun vadeli eğilim bu alanların giderek daha fazla mekanize olmasını sağlamıştır. Son yıllarda, makinelerin işgücüne göre göreceli maliyetinin düşmesi ve pazarların küreselleşmesi ile inşaat sektörü daha büyük miktarlarda yatırıma ihtiyaç duymuş ve büyük ölçekli makine sistemleri ve inşaat tesisi parçalarına erişim daha kolay hale gelmiştir. Bu makineleşme eğilimi, bilgisayar kontrollü inşaat makinelerinin ve esnek imalat kavramlarının endüstriye aşamalı olarak getirilmesi devam edecektir. Donanımın son zamanlardaki gelişimi ve düşük maliyetli bilgisayar ve algılama teknolojisinin mevcudiyeti ile inşaat robotları teknik ve ekonomik olarak endüstride kullanılmaya başlamıştır.

İnşaat robotiği, endüstriyel açıdan önemli bazı işlemleri otomatikleştirmek ve böylece bir insan operatörü kontrol döngüsünden çıkararak bu işlemin maliyetini azaltmak veya makine kontrol

sistemleri aracılığıyla operasyonel verimliliği artırmak için çaba gösterilen gelişmiş bir mekanizasyon (otomasyon) biçimidir [72]. İnşaat işinin doğası gereği, inşaat sektörü için geliştirilen robotların çoğu ya mobil ya da yeri değiştirilebilen sistemlerdir. Robot terminolojisi, araştırma disiplinine bağlı olarak değişebilir. İnşaat robotlarının üç genel kategorisi vardır. Birinci kategori, basitlik için uzaktan kumanda sistemlerini içeren tele-operasyonlu sistemlerdir. İkinci kategori, programlanabilir inşaat makineleri, yerleşik bir insan operatörün çalışmasını artırmak için sensörler ve mekanizmalarla donatılmış çoğu inşaat ekipmanını içerir. Son kategori olan akıllı sistemler, yarı veya tam otonom modda çalışan insansız inşaat robotlarıyla ilgilidir [73].

2.10.7. Maden Robotiği

Madencilik, ticari değer için Dünya'dan mineral kaynakları çıkarma işlemidir. Kırmızı pigment hardalını üretmek için hematit mineralinin çıkarıldığı Paleolitik zamanlara kadar (43 000 yıl öncesine) kadar izlenebilen eski bir insan etkinliğidir. Kömür, sanayi devrimi için kritik olan ve hala modern toplum için önemli olan enerjiyi sağlamıştır ve günümüzde dünyada elektrik enerjisi üretiminin %37'sini oluşturmaktadır. Eski madenlerde taş aletler; ısı su ve daha sonra demir aletler kullanılarak kayalar kırılmıştır. Günümüzün madenleri, büyük dizel ve elektrikle çalışan araçlarla yoğun bir şekilde mekanize edilmekte ve patlayıcılar veya kaya kesme makineleri ile kırılmaktadır.

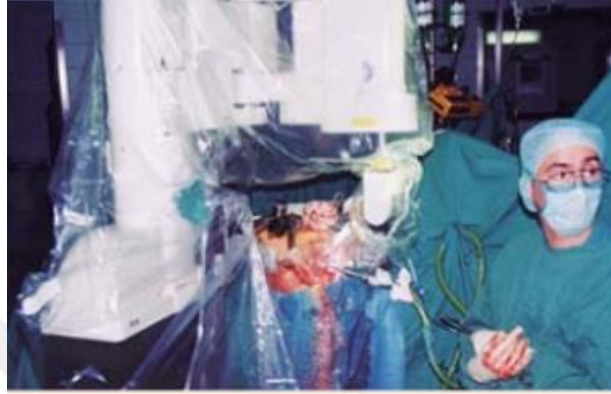
Madencilik zor ve tehlikeli bir iş olması sebebiyle robotik uygulamalara ihtiyaç duyulmaktadır [74]. Madencilik, çok büyük miktarlarda malzemenin uygun maliyetli ve güvenli bir şekilde taşınmasını gerektirir. Yüksek işletme maliyetleri, daha fazla üretkenlik ihtiyacı ve iyileştirilmiş sağlık ve güvenlik sonuçları, robotik için önemli itici güçlerdir [75,76].

Madencilikte karşılaşılan başlıca sorunlar; kapalı alan, yüksek ısı ve nem, uygun olmayan atmosfer ve en önemlisi kötü aydınlatma koşullarıdır. Bu koşullarda mobil yeraltı robotunun katkısı paha biçilmez hale gelir. Atmosferden bağımsız olarak robot bir kazadan hemen sonra harekete geçirilebilir ve kendiliğinden yanma nedeniyle kömür madenlerinde mühürlenmiş panellerin gözetimi sırasında kullanışlı olur. Robotların uygulanması tüm dünyada çeşitli sektörlerde birçok hayat kurtarmıştır ve bu teknoloji gelecekte yeraltı madenlerinde daha sık kullanıldığında sayı katlanarak artabilir [77]. Robot ayrıca madenin 3 boyutlu haritasını çıkarmak ve erişilemeyen yerlerden değerli verileri toplamak için de kullanılabilir [75].

2.10.8. Tıbbi Robotik

İlk çalışmalar 1980'lerin ortalarında olan tıbbi robotiğin büyümesi dikkat çekicidir [78,79]. Stereotaktik beyin cerrahisi, ortopedi, endoskopik cerrahi, mikrocerrahi ve diğer tıbbi alanlara yoğunlaşan bir araştırma topluluğunu içermektedir. Tıbbi robotlar manipülatör tasarımına göre (kinematik, çalıştırma); özerklik düzeyine göre (önceden programlanmışa karşı teleoperasyona

karşı kısıtlı işbirlikçi kontrol); hedeflenen anatomiye veya tekniğe göre (kardiyak, intravasküler, perkütan, laparoskopik, mikrocerrahi); amaçlanan çalışma ortamına göre (tarayıcı içi, geleneksel ameliyathane) olmak üzere farklı şekillerde sınıflandırılabilir. Tıbbi robotikte en önemli ilke; tıbbi robotiğin akademik araştırmacıları, klinisyenleri ve endüstri çalışanları içeren bir ekibe sahip olmasıdır [80,81]. Şekil 2.7’de cerrahi alanında klinik olarak uygulanan ilk robot olan Robodoc sistemi görülmektedir.



Şekil 2.7. Eklem rekonstrüksiyon cerrahisi için klinik olarak uygulanan ilk robot [82,83].

2.10.9. Rehabilitasyon ve Sağlık Hizmetleri Robotiği

Rehabilitasyon robotiğinin tarihi, neredeyse robotiğin tarihi kadar eskidir. 1950'lerin sonlarında başlayan erken robotik, kirli, tehlikeli ve istenmeyen görevler için fabrikalardaki işçilerin yerini alacak büyük manipülatörlere odaklanmıştır. En eski rehabilitasyon robotları protez ve ortez alanında gelmiştir. Case Western Üniversitesi tarafından geliştirilen robotik kolu (1960'lar) ve Rancho Los Amigos Altın kolu (1970'lerin başı), güçlendirilmiş ortezler anlamına gelen yedek mekanik kolların uyarlamaları şeklinde olmuştur [84,85].

Rehabilitasyon robotiği alanı, engelli kişilere gerekli aktivitelerde yardımcı olan veya fiziksel veya bilişsel işlevlerini geliştirmek isteyen kişilere terapi sağlayan robotik sistemler geliştirir [86]. Kişisel robotlar, robotik terapi, akıllı protezler, akıllı yataklar, akıllı evler ve tele-rehabilitasyon hizmetleri geliştirmeye yönelik programlar son yıllarda hız kazanmıştır ve insanların kaliteli yaşamasına imkan vermek için sağlık hizmetlerinde yeni gelişmelerin yaşanması sürecini hızlandırmıştır. Rehabilitasyon robotiği alanı genellikle terapi ve yardım robotları kategorilerine ayrılır. Ek olarak rehabilitasyon robotiği, yapay uzuv (protez) geliştirme, fonksiyonel nöral stimülasyon ve aktiviteler sırasında insanların teşhisi ve izlenmesi için gerekli teknolojiyi içerir. Terapi robotları terapiyi alan engelli kişi ve robotla etkileşimi kuran ve izleyen terapist olmak üzere genellikle aynı anda en az iki ana kullanıcıya sahiptir. Robotik yardımdan yararlanan terapi türleri, otizmlı çocuklar için iletişim sağlayan ve serebral palsili veya diğer gelişimsel engelleri olan çocuklar için keşif (eğitim) sağlayan üst ve alt ekstremiteler hareket terapisi. Robot, birkaç nedenden dolayı gerçek uygulamalı müdahale için bir fizyoterapistin iyi bir alternatif olabilir ve

terapiyi uzun süre boyunca yorulmadan düzgün bir şekilde uygulayabilir. Robotun sensörleri, hasta tarafından gerçekleştirilen eylemleri ölçebilir ve iyileştirebilir. [86,87].

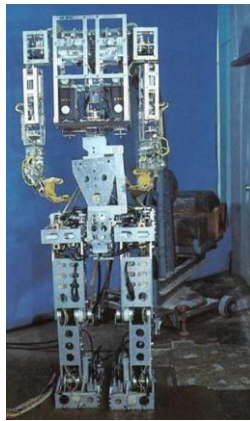
Yardımcı robotlar genellikle manipülasyon, hareketlilik veya biliş odaklanmalarına göre gruplandırılır. Manipülasyon yardımcıları ayrıca sabit platformlu, taşınabilir platformlu ve mobil özerk tiplerde sınıflandırılır. Sabit platformlu robotlar, mutfakta, masaüstünde veya engelli kişiye ait yatağın yanında belirli işlevleri üstlenir. Taşınabilir tipler, bir kapıyı açarken olduğu gibi nesneleri tutmak ve hareket ettirmek ve diğer cihaz ve ekipmanlarla etkileşime geçmek için elektrikli tekerlekli sandalyeye bağlı manipülatör kollarıdır. Mobil otonom robotlar, evde veya işyerinde manipülasyon ve diğer görevleri yerine getirmek için sesle veya başka yollarla kontrol edilebilir. [88–90].

2.11. Robotikte Karşılaşılan Temel Problemler

2.11.1. Yürüme

Bacaklı (İnsansı) robotlar alanındaki ilk çalışmalar 1970'lerde Kato ve Vukobratovic tarafından gerçekleştirilmiştir [91,92]. Bu çalışmada ilgili deneysel sistemlerin tasarımı ile gerçekleştirilmiştir. Japonya'da ilk antropomorfik robot olan WABOT-1, 1973 yılında Kato ve arkadaşları tarafından Waseda Üniversitesi'nde sergilenmiştir. Çok basit bir kontrol şeması kullanılan WABOT 1, statik dengede birkaç yavaş adımı gerçekleştirebilmiştir. Buna paralel olarak, M. Vukobratovic ve arkadaşları, fonksiyonel rehabilitasyonun yarattığı problemlerle yakından ilgilenmiştir.

Yugoslavya'daki Mihailo Puppin Enstitüsünde, ilk aktif dış iskeletler ve Belgrad'ın eli gibi diğer birkaç cihaz tasarlanmıştır, o zamandan beri yaygın olarak kullanılan yürüme dengesi analizleri olmuştur [93]. Buradaki yaklaşım, klasik bir statik denge kriterini genişletmek için dinamik anahtar kullanmak olmuştur. İlk bacaklı modern robot olan WABOT 1 Şekil 2.8'de görülmektedir [91].



Şekil 2.8. İlk insan benzeri modern robot [91]

Daha sonra yeni gelişmeler ABD’de yaşanmıştır. R. McGhee'nin 1960'larda Güney Kalifornia Üniversitesi'nde, ardından 1970'lerde Ohio Devlet Üniversitesinde ilk bilgisayar kontrollü yürüme makinesiyle sonuçlanan çalışma yapılmıştır. Sonrasında M. Raibert robot dinamiği alanında Carnegie Mellon Üniversitesi'nde çalışmaya başlamıştır. Bacaklı robotlarla ilgili araştırmalar için üçüncü önemli dönem 1990'ların başında başlamıştır. Tamamen pasif mekanik sistemleri inceleme fikrine McGeer öncülük etmiştir [94,95]. Bu makalede McGeer, çok basit bir sistem sınıfı için doğal döngüsel davranış kavramını tanıtmıştır. Ancak burada önemli olan nokta McGeer'in Poincare haritalarını kullanarak bu tür sistemlerin yörünge stabilitesi açısından analizini robotikçiler için kullanışlı hale getirmesidir. 20. yüzyıl yoğun teknolojik faaliyetlerin olduğu bir dönem olmuştur. Endüstriyel atılımlar dünyaya gerçek insansı robotları inşa etmenin mümkün olduğunu göstermiştir [96]. Japonya'da, ilk insansı robot P2, 1996 yılında Honda tarafından sergilenmiştir ve ardından birkaç örneği daha geliştirilmiştir [97]. Şu anda, en etkileyici teknik başarılar hala endüstriyel şirketler tarafından gerçekleştirilmektedir. ASIMO (Honda) [98], QRIO (Sony) [99], HRP (Kawada) [100] ve NAO (Aldebaran Robotics) [101] diğerlerinin yanı sıra bugünün başlıca örnekleridir. Buna paralel olarak, özellikle eğlence amaçlı küçük insansı robot pazarının son on yılda istikrarlı bir şekilde büyüdüğü görülmektedir.

2.11.2. Kavrama

Robotlara değişen geometrik ve fiziksel özelliklere sahip nesneleri kavrama yeteneği vermek için mekanik eller geliştirilmiştir. Manipülasyon için tasarlanan ilk robotik el, üç eklemlili parmağa sahip Salisbury eli olmuştur [102]. Birçok robot için vazgeçilmez bir işlem olan kavrama, yıllardır yoğun araştırmalara konu olmuştur [3,30,103,104]. Birçok uygulama senaryosunda, özellikle belirsiz veya yapılandırılmamış ortamlarda etkin bir şekilde çalışabilmesi için robotik bir kavrayıcı, çeşitli pozlardaki çeşitli boyut ve şekillerde nesneleri kavrama yeteneğine sahip olmalıdır [5,105–107].

Nesneleri kavrama, taşıma ve yerleştirme, robotlar ve robot manipülatörler için temel yetenekler ve ortak işlemlerden biridir. Kavrayıcılar (grippers), robotların en temel bileşenleridir ve robotlar ile ortamlar/kavranan nesneler arasındaki mekanik arabirimin yanı sıra uç işlevci olarak hizmet ettikleri için birçok manipülasyon görevinde önemli rol oynamaktadır. Kavrayıcılar, manipülasyonlarda kavranan nesnelerle geçici temas sağlar. Güvenli kavrama, yalnızca nesnelerle temas etmeyi değil, aynı zamanda nesneler alınırken ve yerleştirilirken olası kayma ve hasar riskinin önlenmesini de gerektirir. Çok çeşitli şekil, boyut ve malzemeye sahip nesneler için güvenli kavrama sağlamak için çeşitli sensörler ve kontrol stratejileri de gereklidir. Teknolojilerin gelişmesi, nüfusun yaşlanmasının neden olduğu işgücü sıkıntısı ve yüksek otomasyon derecesi gereksinimleri ile birlikte birçok alanda yerini bulacaktır. Robotlar için kol ucu araçları olarak,

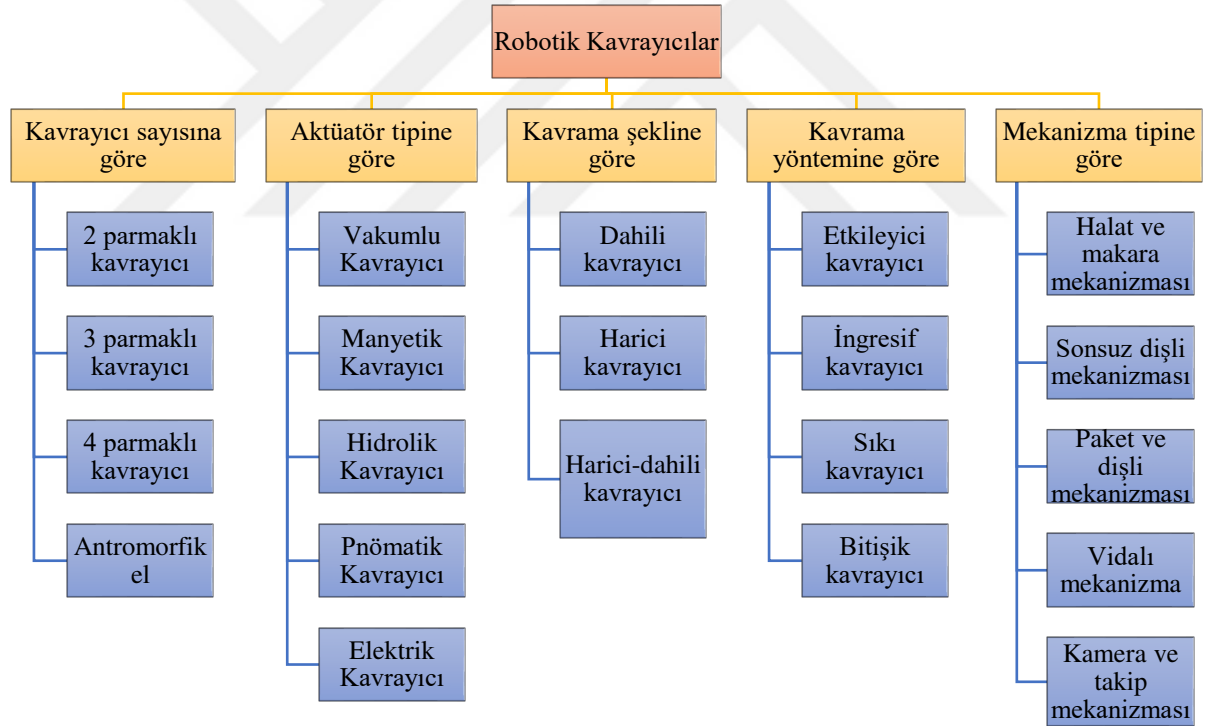
kavrayıcılar robotların elleri olarak görülebilir, neredeyse tüm otomatik manipülasyonlar doğrudan robotik kavrayıcılar tarafından gerçekleştirilir [7,10,12,17,108–110].

Robotlar ve robot manipülatörler; sıkıcı, tekrarlayan, kirli veya tehlikeli işlerde insanlara yardımcı olmak veya onların yerini almak için geliştirilmiştir [111]. Farklı şekil, boyut, malzeme ve yüzey özelliklerine sahip nesneleri alıp yerleştirmek gibi insanlara basit görünen görevler robotlar için zor olabilir [112]. Kavrayıcı; iş parçalarını kavramak, taşımak ve yerleştirmek için ekipmana monte edilmiş bir kol ucu aracıdır. Elektrik, pnömatik veya diğer türden güçle çalışan parmakları açıp kapatarak, kavrayıcı iş parçalarını kavrar ve serbest bırakır. Robotik kavrayıcılar, nesneleri ve çevreyi kavramakla etkileşime girmekle görevlendirilir [111]. Robotik kavrayıcılar, robotların alma ve yerleştirme (pick and place) işlemlerini gerçekleştirmek için robot elleri olarak hizmet ettikleri için robotik manipülatörlerin temel bileşenleridir [113]. Robotik kavrayıcıların esneklik açısından insan işçilerle rekabet etmesi daha zor hale gelmektedir [114]. Robot endüstrilerinde otomasyonun evrimi ile birlikte, kavrayıcılar ve kavrama, robotik araştırma topluluğunda önemli trend konuları haline gelmiştir. İlk kavrayıcı modüller, 30 yıl içerisinde standart ürünler olarak geliştirilmiştir. 1975'ten bu yana kavrayıcılar, çok sayıda farklı gereksinimler, çeşitli iş parçalarının ve iyi uyarlanmış ve güvenilir sistem talebi nedeniyle hızla geliştirilmiştir [115]. İlk kavrayıcılar endüstriyel uygulamalar için geliştirilmiş [111], ardından robotik kavrayıcıların uygulamaları günlük hayatımızdaki diğer alanlara da yayılmıştır. Kavrayıcı, otomatikleştirilmiş ve/veya robotize edilmiş bir çözüm kullanmakla pratik başarısına büyük katkı sağladığından, uygun bir tasarım temel öneme sahiptir. Uygun bir kavrayıcı tasarımı ayrıca genel robot sistem montajını basitleştirebilir, genel sistem güvenilirliğini artırabilir ve sistemin uygulama maliyetini azaltabilir. Başarılı bir kavrama işlemi için kavrayıcı tasarımı ve seçimi çok önemlidir. Kavrayıcılar farklı sınıflandırma kriterlerine göre farklı kategorilere ayrılabilir. Robotik kavrayıcıdan bahsederken, etkili kavrayıcılar, ingresif kavrayıcılar, kısıtlayıcı kavrayıcılar ve bitişik kavrayıcılar olmak üzere dört genel robot kavrayıcı kategorisi bulunmaktadır [115]. Şekli ve boyutu değişen nesneleri kavrama sırasında olası kayma ve hasar olmadan kavrayabilmek ve gerekli sıralı işlemleri yapabilmek için çeşitli sensörlere ve kavrama kontrol stratejilerine de ihtiyaç duyulmaktadır [116]. Kavrayıcılara entegre edilen sensörler, sıcaklığı, kuvveti, konumu, kaymayı ölçebilir. Kontrol teknolojisi ile birlikte sensörler, kavrayıcıların çalışmasını daha akıllı hale getirebilir [117].

2.11.3. Kavrayıcıların Sınıflandırılması

Robotik kavrayıcılar, Şekil 2.9'da gösterildiği gibi farklı sınıflandırma stratejilerine göre farklı kategorilere ayrılabilir [118]. Kavrayıcıların sınıflandırılması, parmak sayısı, çalıştırma türleri, kavrama modları, mekanizma türleri ve fiziksel kavrama ilkeleri göz önünde bulundurularak yapılmaktadır.

Parmak sayısına göre robotik kavrayıcılar; 2 parmaklı kavrayıcı, 3 parmaklı kavrayıcı, 4 parmaklı kavrayıcı ve antropomorfik el olarak dört ana kategoriye ayrılır. Parmak sayısı, kavranan nesnelerle etkileşime girerken kavrama işleminin temas yüzeylerine karar verir. 2 parmaklı kavrayıcı, üreticilerin kataloglarında en yaygın görülen kavrayıcı tipidir [119]. İnsanların nesneleri kavramasının %60 ile %70'i sadece iki parmakla gerçekleştirilir ve bu nedenle iki parmakla kavrama gerçek dünya uygulamalarında oldukça popülerdir [120,121]. Çok parmaklı kavrayıcılar güvenli tutuş sağlayabilir ancak kontrol stratejileri daha karmaşıktır. Çalıştırma türlerine göre robotik kavrayıcılar, vakumlu, manyetik, hidrolik, pnömatik ve elektrikli tutucular olarak sınıflandırılabilir. Vakum kavrayıcılar, kavrayıcı plaka ile nesne yüzeyi arasında yüksek hızlı bir hava akışı üretmek ve böylece vakum oluşturmak için Bernoulli Prensibi üzerinde çalışır [122]. Pnömatik kavrayıcılar ve elektrikli kavrayıcılar en popüler kullanılan kavrayıcılardır. Pnömatik kavrayıcılar yüksek kavrama kuvveti sağlayabilir ve elektrikli kavrayıcılar kolay ve hassas bir şekilde kontrol edilebilir. Hidrolik sistemlerde yardımcı yapı gereksinimi nedeniyle [118], hidrolik kavrayıcılar, ağır vücut ağırlığı dezavantajının yanı sıra güçlü kavrama sunar.



Şekil 2.9. Farklı sınıflandırma stratejilerine göre farklı robotik kavrayıcılar [118]

Kavrama modlarına göre dahili, harici ve harici-dahili kavrayıcılar mevcuttur. Kavrayıcılar ve kavrama modları, kavranan nesnelerin yapısına göre seçilebilir. Mekanizma tiplerine göre kavrayıcılar; vidalı mekanizma [123], paket ve pinyon mekanizma [124], kamera ve takipçi mekanizma [125], halat ve makara mekanizma [126] ve sonsuz dişli mekanizma [127] olmak üzere beş ana kategoriye ayrılır. Fiziksel çalışma prensibine göre kavrayıcılar; etkileyici, saldırgan,

kısıtlayıcı ve bitişik olmak üzere dört ana gruba ayrılır [115]. Kavrayıcı sınıflandırma stratejileri için Monkman ve arkadaşları tarafından önerilen fiziksel çalışma prensibine dayalı sınıflandırma yöntemi, araştırmacılar tarafından yaygın olarak kabul edilmektedir [115]. Etki grubu ile nesnelere iki veya daha fazla yönden doğrudan bir mekanik kuvvet uygulanır. İngresif grup ile nesnenin kavranması nesne yüzeyinin nüfuz etmesiyle sağlanır. Sıkı kavrayıcı grup ile tek bir yönde bağlayıcı bir kuvvet uygulanır. Bitişik grup ile tek bir yönde bir tutma kuvveti üretmek için doğrudan bir sözleşmenin gerekli olduğu, etkili olmayan yöntemlerdir.

2.11.4. Robotik Kavrama

Nesne kavrama, herhangi bir üretim otomasyonunun gerekliliklerinden biridir. Robotik kavrama, kapsamlı bir işlem olup kavranan nesnenin özelliklerinden etkilenir. Nesneleri kavrama ve serbest bırakmanın yanı sıra robotun çevreyi algılaması ve çevre ile etkileşime girmesi beklenir. Nesnenin konumu ve pozunu hakkında doğru bilgi, güvenli ve istikrarlı kavrama için önemlidir. Önceki manipülasyon, nesne lokalizasyonu, nesne poz tahmini ve segmentasyon yöntemlerini içerir. Nesne segmentasyonu, hedef nesneleri arka plandan çıkarır. Nesne lokalizasyonu, kavrayıcının doğru yaklaşması için nesne konumu bilgisi sağlar. 6 boyutlu (6B) poz tahmini, doğru kavrama hareketini elde etmeye yardımcı olmak için bir referansa göre hedef nesnenin ötelenmesini ve dönüşünü hesaplar.

Kavrama Sürecinin Aşamaları

Belirsizlikleri azaltmak için, kavrama süreci birkaç standart aşamaya bölünebilir [128]. 2 parmaklı mekanik kavrayıcının kavrama sürecinin 7 standart aşaması şunlardır:

- Nesneye yaklaşma: Kavrayıcı açık ve yakalanan hedefe yakın konumlandırılır.
- Temas kurma: Kavrayıcı çalıştırılır ve hedef yüzeye temas eder.
- Tutma kuvvetinin artırılması: Sabit ve güvenli kavrama sağlamak için uygun kuvvet yüklenir.
- Nesneyi sağlama alma: Kavranan hedefin dinamik dengesini sağlamak için yeterli kuvvet uygulanırken, kavrama kuvvetinin artması durur.
- Nesnenin kaldırılması: Nesne kaldırılır ve istenen konuma hareket ettirilir.
- Nesneyi serbest bırakmak: Kavrama kuvveti kaldırılır ve tutulan nesne yerçekimi yardımıyla serbest bırakılır.
- Kavramanın izlenmesi: Kuvvet sensörleri, temas sensörleri, yapışma-kayma sensörleri, tork sensörleri gibi çeşitli sensörler kullanılarak temas ve kavramanın etkinliği izlenmelidir.

Kavrama Stratejileri

Kavrama stratejileri; kavrama planlaması ve kavrama modelleri gibi farklı uygulamalarda farklı olabilir. Kavrama stratejileri, bir nesneyi kavrama yaklaşımının tanımının temeli veya başlangıç noktasıdır. Gerçek dünya uygulamalarında kavrama işlemlerinde geometrik ve fiziksel özellikler, mekanik özellikler ve operasyonel özellikler göz önünde bulundurulmalıdır [129]. Kavranan nesnenin özelliklerine ve kullanılan kavrayıcının konfigürasyonuna göre uygulamalarda kombinasyon stratejileri ve modelleri uygulanabilir. Robotik kavramanın performansı; kavrama sağlamlığı, güvenilirliği ve kararlılığına göre değerlendirilebilir [130]. Sağlamlık, bir kavrama sisteminin, ilk kararlı konfigürasyonunu uyarlamadan değişime direnme yeteneğini ölçmek için kullanılabilir. Güvenilirlik, kavrama güvenliğini ve esnekliğini değerlendirmek için kullanılabilir. Kararlılık, kavrayıcının dengeli kavrama sağlama yeteneğini değerlendirmek için kullanılabilir [131].

2.11.5. Kavrama Yöntemleri

Kavrama yöntemleri mekanik yöntemler, pnömatik yöntemler, bağlı yöntemler ve elektromanyetik yöntemler olmak üzere dört gruba ayrılır [132]. Kavrama işlemlerinin çoğu, paralel kavrayıcılar ve çok parmaklı kavrayıcılar kullanılarak iki veya daha fazla yönden mekanik kuvvetle gerçekleştirilir. Parmaklardaki veya parmak uçlarındaki dokunma sensörler veya akıllı malzemeler, kavrayıcıların gerçek zamanlı hassas kontrolü için uygulanabilen kavrama kuvvetlerinin ölçülmesini sağlar. *Mekanik yöntemlerin* avantajları, farklı nesne şekilleri ve boyutları için esnek kavramalarının yanı sıra geri beslemeli kavrama kuvvetidir ancak aynı zamanda karmaşık kontrolde dezavantajlar da getirirler [111].

Pnömatik yöntemler, nesneleri vakumlu kavrayıcılar veya Bernoulli kavrayıcılar kullanarak kavrayabilir. Vakum kavrayıcılar, vantuzlarla güçlü bir içe doğru emme kuvvetini harekete geçirerek nesneleri kavrar [113]. Bernoulli kavrayıcılar her zaman radyal hava akış başlıklarından oluşur. Bu başlıklar, kendi plakası ile nesne yüzeyi arasında yüksek hızlı hava akışı oluşturabilir ve nesne yüzeyindeki basınç, aşağıda yukarıdan daha yüksek olacaktır. Alt ve üst plakalar arasındaki basınç farkından dolayı yukarıya doğru bir kaldırma kuvveti oluşturulabilir, böylece kavrayıcılar cismi kaldırabilir. Pnömatik yöntemler, yarı iletken endüstrisinde ve gıda endüstrisinde büyük, düz, pürüzsüz nesnelerin ve ayrıca küçük düz malzemelerin kaldırılmasında avantajlara sahiptir.

Bağlı yöntemler, kimyasal yöntemler, fiziksel yöntemler veya yüzey gerilimi kullanılarak geçici yüzey yapıştırma yoluyla nesneleri kavrar. Bağlı yöntemlerinin kavrama kuvvetleri, kavrayıcıların birleştirme kapasitesi veya yüzey gerilimi tarafından oluşturulur. Tut ve yerleştir işlemlerinde bağlı yöntemlerin maksimum toplama kuvveti ve minimum serbest bırakma kuvveti, belirli bir nesne için tanımlanmalıdır.

Elektromanyetik yöntemler, elektrik alan veya manyetik alan kullanarak iç malzemelerin viskozitesini değiştirerek ve esnek dış yüzey kullanarak kaldırma kuvveti oluşturarak gerçekleştirilir. Kontrol ekipmanının açılıp kapatılmasıyla elektrik alanı veya manyetik alan etkinleştirilebilir. Yumuşak kavrayıcıların çoğu, elektromanyetik kavrama yöntemlerine dayanmaktadır, bu yöntem, kavranan nesnenin değişken şekil ve boyutlarına, duyuşal geri bildirim gerektirmeden uyarlanabilir güvenilir kavrama sağlayabilir.

2.11.6. Sensör Tabanlı Kontrol Stratejileri

İnsan eli, hem sensörleri hem de taşıma aletini entegre eden bileşik kavrama sistemi olarak düşünülebilir. İnsan eli, beynin kontrolü altındaki duyuşal geri bildirim ve önbilginin yardımıyla manipölasyonlar için istikrarlı, uyarlanabilir ve esnek kavrama ve diğer olağanüstü yetenekleri sağlayabilir. Tüm robotlar; dokunma sensörleri, görsel sensörleri, işitme sensörleri vb. dahil olmak üzere çeşitli sensörlerle etkileşime girmeli, değişken ve karmaşık fiziksel ortamları arayüz oluşturmalı [99] ve kapalı döngü kontrol politikalarının geliştirilmesini ve uygulanmasını kolaylaştırmalıdır [133]. Robotik kavrayıcılar, çeşitli nesneleri dokunma ve görsel algı ile kavramada çoklu sensörler kullanarak insan elinin avantajlı yeteneklerini taklit etmeye çalışırlar. Birden fazla sensörü entegre ederek, kavrayıcılar nesneyi yalnızca robotların bir kol sonu aracı olarak kavramakla kalmaz, aynı zamanda kavrayıcı ve nesne arasındaki bilgileri analiz edebilir. Bu arada, çevrimiçi karar verme stratejisi de füzyon duyuşal verilere dayalı olarak yürütülebilir. Bu nedenle, kavrayıcılar bir bilgi kaynağı olarak görülebilir. Sensör entegrasyonu, uyarlanabilirlik, dokunma hissi, görme keskinliği, etkileşim ve stabilite gibi kavrayıcı niteliklerde fayda sağlayacak güçlü işlevsellik sağlayabilir. İlgili tahrik ve kontrol teknolojisi ile birlikte sensörler, kavrayıcıyı daha akıllı hale getirebilir [117].

Anahtarlama Sensör Tabanlı Kontrol Stratejileri

Anahtarlama sensörleri hareketin belirli bir konuma ulaşıp ulaşılmadığını belirlemek veya mekanik sistemler için bir tetikleme sinyali sağlamak için kullanılır. Anahtarlama sensörleri, otuz yılı aşkın bir süredir mekanik tutucularda yaygın olarak kullanılmaktadır [117]. Gerçek dünya uygulamalarında, kavrayıcı açık, kavrayıcı kapalı, nesne tutulmuş gibi kavrayıcı konum durumunu izlemek için anahtarlama sensörleri kullanılabilir veya mekanizmaların daha fazla eylemini etkinleştirmek için ikili bir açma-kapama sinyali sağlayabilir. Mesafe anahtarları, manyetik (reed) anahtarları, dokunmatik anahtarlar ve hall anahtarı sensörleri robotik kavrayıcılarda kullanılan dört anahtarlama sensörü türüdür.

Mesafe (proximity) anahtarları; algılama, otomasyon ve kontrol için kritik temassız sensörlerdir. Mesafe anahtarları, ayrı bir ikili çıkış sinyali ileterek, oluşturulan algılama alanlarının

aktif aralığı içinde gelen hedefe yanıt verir. Sinyal birleştirme, kapasitif, endüktif, akışkan, akustik olarak yansıtan, optik olarak yansıtan, optik iletim araçlarıyla gerçekleştirilebilir [115].

Manyetik anahtarlar; uygulanan bir manyetik alan tarafından çalıştırılan elektrik anahtarlarıdır. Kontaklar normalde kapalı olabilir ve bir manyetik alan varken açılır veya bir manyetik alan uygulanırken normal olarak açılır ve kapanır. Mıknatıs anahtardan çıkarıldığında, manyetik anahtar orijinal konumuna geri döner [134].

Dokunmatik anahtarlar; çalışması için yalnızca bir nesnenin dokunması gereken anahtar türleridir. Dokunmaya duyarlı robotik kavrayıcı, nesneleri yer değiştirme ölçümlerine ve/veya diğer ölçümlere göre tanımlamak için yapılandırılmış bir işlemci ve/veya bellek içerir. Dokunmatik anahtarlar, belirli bir sinyal için eşikleri kolayca kaydedebilir ve diğer eylemler için sinyal sağlayabilir. Örneğin, dokunmatik anahtarlar, emme kafası ve hedef arasındaki temas tarafından etkinleştirildiğinde, kavramayı etkinleştirmek için vakumu değiştirmek için kontrol sinyali üretilir [115].

Hall anahtarı sensörü; hall etkilerine dayalı olarak bir manyetik alana yanıt olarak çıkış voltajını değiştiren bir tür manyetizma algılama anahtarıdır. Hall anahtarı sensörleri her zaman mesafe anahtarlama için kullanılır. Kampmann ve arkadaşları, dokunma algılama elemanları ile donatılmış endüstriyel bir derin deniz manipülatörünün çene kavrayıcısı için bir damla değiştirme geliştirmiştir [135]. Sensör yüzeyi ile çeşitli şekillerdeki nesneler arasında temas oluşturularak elde edilen dokunma geri bildirim, Hall-etki sensör çenesi kullanılarak çene modülü için gösterilir. Hall etkisi sensörleri, su altında farklı şekillerde nesneleri tutarken dokunma algılama olmadan dokunma geri bildirim ve endüstriyel çene kavrayıcıları kullanarak nesne temasını tanıyabilir.

Dokunma Sensörü Tabanlı Kontrol Stratejileri

Dokunma sensörleri, kendisi ile nesneler arasındaki fiziksel temaslar yoluyla nesnelerin dokunsal özelliklerini ölçebilen algılama bileşenleri olarak tanımlanır. Dokunma sensörleri fiziksel temasa tepki verir, bu nedenle robotun kavrayıcısı ile kavranan bir nesne arasında temas kurulduğu anda bilgi sağlayarak görsel sistemleri tamamlar. Nesnenin mekanik özelliklerinin yalnızca görme ile belirlenemeyeceği durumlarda, görüş yeterli olmadığından dokunma sensörlerinden faydalanılabilir. İlkelere göre, dokunma sensörleri oluşturmak için birçok form ve dizi konfigürasyonu mevcuttur. En yaygın dokunma sensörleri; kapasitif, dirençli, optik (kızılötesi veya görünür ışık), ultrasonik, manyetik ve piezo-elektrik temellidir. Dokunma sensörleri; basınç, kuvvet, tork, kayma, titreşim, sıcaklık ve nem gibi mekanik özellikleri de ölçebilir. Robotik manipülasyonda, temasın temas kuvveti ve torkları gibi mekanik özellikleri incelenir.

Kuvvet geri beslemesi, kavrayıcılar ve iş parçaları arasındaki fiziksel etkileşimi içeren hemen hemen tüm manipülasyon görevlerinin başarısı için temel bir gerekliliktir [130]. Bir kavrayıcı, temas kuvvetini algılamak için parmak uçlarında dokunma sensörü dizileriyle

donatılabilir. Genel olarak, robotik manipölasyon kontrolü, kavrayıcılar ve hedefler arasındaki temaslar sırasında meydana gelen etkileşimler hakkında kapalı döngü bilgisi sunabilen duyuşal geri bildirimler gerektirir. Kuvvet-tork ölçümü, otomatikleştirilmiş montaj işlemlerinin kontrolü ve gerçek zamanlı izlenmesi için çok önemli bir sinyaldir [115]. Dokuma sensörleri, hassas geri besleme kontrolü dahil olmak üzere birçok amaç için son yıllarda robotik kavramada uygulanmıştır. Robotik kavrayıcıların kontrolünü daha hassas ve daha akıllı hale getirmek için hala üstesinden gelinmesi gereken zorluklar bulunmaktadır.

Ölçüm Sensörü Tabanlı Kontrol Stratejileri

Ölçüm sensörleri, robot kontrolörünün program içinden hedef konumları güncellemesini sağlamak için mesafeyi veya kavrayıcının parmak aralığını belirlemek için daima kavranan nesnenin boyutlarını ölçmek için kullanılır [115,117]. Mesafe ölçümü için akustik sensörler, mikrodalga ve lazer üçgenleme sensörleri, hall etkisi sensörleri, ivme sensörleri, kuvvet-tork sensörleri ve kayma sensörleri kullanılabilir. Akustik sensörler ve lazer üçgenleme sensörleri, sırasıyla ultrasonik ölçüm mesafesi teknolojisi ve trigonometrik ilkeye dayalı olarak mesafeyi ölçer. Hall efekt sensörü, manyetik alana tepki olarak çıkış voltajını deęiştiren bir dönüştürücüdür. Hall efekt sensörleri, konumlandırma, hız algılama, yakınlık deęiştirme ve akım algılama uygulamaları için yaygın olarak kullanılır.

Görsel Sensör Tabanlı Kontrol Stratejileri

Görsel sensörler, optik görüntüleri elektronik sinyallere dönüştüren cihazlardır. Daha çok dijital kameralarda, kamera modüllerinde ve dięer görüntüleme cihazlarında kullanılmaktadır. Görsel sensörler, robotik manipölasyonların nesneleri konumlandırması, ortamdaki kavrama noktalarını belirlemesi ve çevresindeki engelleri belirlemesi için en uygun temassız sensörlerdir [130]. Görsel sensörler ayrıca nesne ile teması algılamak, nesnelerin 3B (üç boyutlu) yapılarını algılamak ve kavrayıcılar ile kavranan nesneler arasındaki ilişkiyi izleyerek görsel geri bildirim sağlamak için kullanılır [131]. Bu nedenle, görsel sensörleri kavrayıcılara entegre etmek, kavrama ve taşıma sürecini kontrol etmeyi mümkün kılabilir [117]. Hem monoküler (monocular) kamera hem de stereo kamera görüş sistemleri, robotik kavrayıcılar ve robot kavrama uygulamalarında bulunmaktadır.

Monoküler kameralar, tek kamera sistemlerini kullanarak yalnızca 2 boyutlu görüntüler üretebilir. Shaw ve Cheng, robot manipölasyonları için bir görüş sistemi ile entegre akıllı bir kavrayıcı tasarlayıp geliştirmişlerdir [136]. Geliştirilen görüş sistemi, robot manipölatörün kavrama eylemini gerçekleştirmesinde nesneyi tanımak ve bulmak için yalnızca tek bir kamera kullanmıştır. Nesnenin konumu, tek kamera tarafından iki farklı konumda yakalanan görüntüler kullanılarak belirlenebilir. Stereo kameralar; her lens için ayrı bir görüntü sensörüne sahip iki veya daha fazla

lensli kamera türleri olup 3 boyutlu görüntüler üretebilir, çevre ve kavranan nesnelerle ilgili bilgi sağlayabilir. Robotların çoğu stereo görüş sistemi ile monte edilmiştir [130]. 3B bilgi edinme için robotik manipülasyon görevlerinde kinect sensörleri, dürbün kameraları, uçuş süresi kameraları ve kavranan nesne tanıma her zaman uygulanır [137,138].



3. DERİN PEKİŞTİRMELİ ÖĞRENME

Bu bölümde, uygulamalarda kullanılan yöntemlerin daha iyi anlaşılması için YSA, derin öğrenme ve DPÖ yöntemlerine ait detaylı bilgiler sunulacaktır.

3.1. Makine Öğrenmesine Genel Bir Bakış

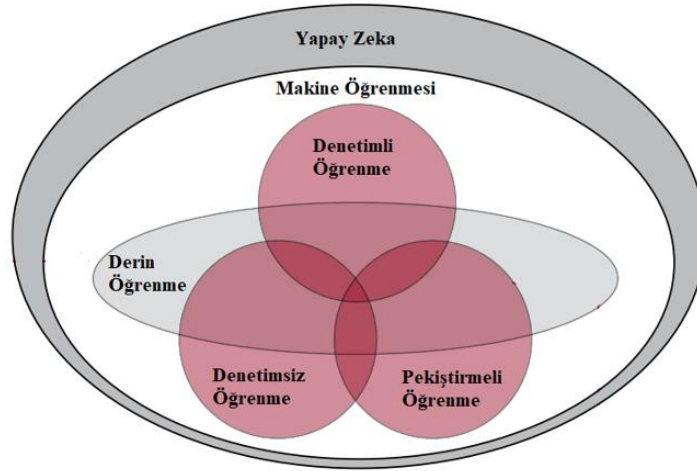
Yapay zekâ, gelecekte endüstride yaşanacak yeni köklü değişikliklerin önünü açmıştır [139]. Elektriğin hayatımıza girmesiyle her alanda getirdiği kolaylıklar ve sunduğu imkânlar gibi yapay zekânın da birçok alanda çığır açıcı gelişmeler yaşatmaya başlayacağı öngörülmektedir.

Yapay zekâ ile ilgili neredeyse her gün bilimsel makale çalışmaları yapılmaktadır. Özellikle son yıllarda teknolojiye yeni yönelimlerin baş mimarı olmaya başlamıştır. Güncel tanımı “normalde insan tarafından gerçekleştirilen görevlerin düşünebilen sistemlerle otomatik hale getirilmesi” olarak yapılabilmektedir. Yapay zekâ serüveni kavramsal olarak ilk defa 1956 yılında John McCarthy tarafından düzenlenen bir konferansla başlamıştır [140]. “Makineler düşünebilir mi” sorusu ise daha önce gündeme gelmiştir [141]. Enigma makinesi II. Dünya savaşı sonunda askeri iletişimde gizli mesajların şifrlenmesi ve tekrar çözülmesi amacı ile geliştirilmiştir. Alan Turing enigma kodlarını çözen bir yapay zekâ sistemi kurmuştur. Enigma kodunun insan tarafından çözülmesi oldukça zor olması ve zaman alması sebebiyle saatler içinde sonuç alınabilecek bir yapay zekâ makinesi ortaya çıkmıştır. Sonraki yıllarda yapay zekânın gelişimine katkıda bulunan önemli gelişmeler aşağıdaki gibi sıralanabilir.

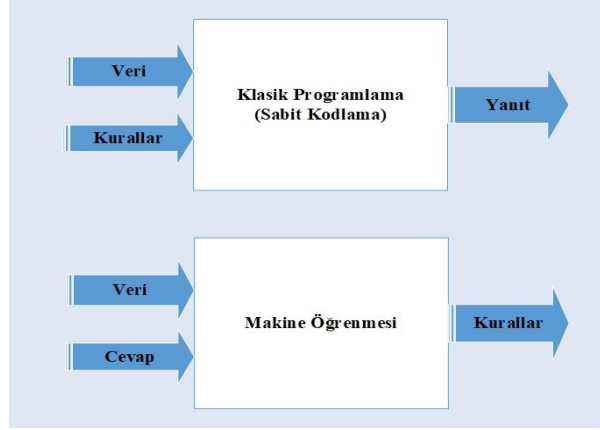
- Sinir Ağları – Warren McCulloch ve Walter Pitts, 1943 [142]
- Turing Testi – Alan Turing, 1950 [141]
- Yapay Zekânın Doğuşu – John McCarthy, 1956 [143]
- Algılayıcı (Perceptron) – Frank Rosenblatt, 1957 [144]
- Neocognitron – Kunihiko Fukushima, 1980 [145]
- Geriye Yayılım – Geoffrey Hinton, 1986 [146]
- Özyinelemeli Sinir Ağları – Jeffrey Elman, 1990 [147]
- LeNet – Yann LeCun, 1998 [148]
- Derin İnanç Ağları – Geoffrey Hinton ve ekibi, 2006 [149]
- ImageNet – Fei Fei Li, 2009 [150]
- AlexNet – Alex Krizhevsky, 2012 [151]
- GAN – Ian Goodfellow, 2014 [152]
- AlphaGo – Google DeepMind, 2016 [153]
- Kapsül Ağları – Geoffrey Hinton, 2017 [154]
- GPT-3 – OpenAI, 2020 [155]

İnsan için çözümü sezgisel olarak kolay olan bazı problemler, bilgisayarlar için zor olabilmektedir. Kedi ile köpeğin ayırt edilmesi, buluşmaya gecikince arkadaşınızın size kızgın olup olmadığını gösteren yüz ifadesinin yorumlanması, herhangi bir seminer ya da toplantıda alınan notların dilinizi anlamayan farklı bir dildeki arkadaşınızın anlaması için dönüştürülmesi gibi durumlar örnek verilebilir. İnsanda bu tür görevler sezgisel olarak yapılırken, bilgisayarların yapabilmesi için programlanması gerekmektedir. İlk yapay zekâ sistemlerinde genellikle sabit kodlama ile satranç oynama gibi görevler ele alınmıştır. Yapay zekânın ilk yıllarında birçok araştırmacı, yapay zekânın sabit kodlama kuralları ile elde edilebileceğine inandıkları için iyi tanımlanmış mantıksal problemlerde olumlu sonuçlar almışlardır. Ancak imge tanıma, nesne tespiti, nesne bölütleme, Doğal Dil İşleme (DDİ) gibi karmaşık görevlerde yetersiz kaldığı görülmüştür. Karmaşık olarak bilinen bu görevlere, makine öğrenmesi ve derin öğrenme gibi güncel yapay zekâ yaklaşımları ile çözümler getirilmiştir [156].

Yapay zekâ, makine öğrenmesi ve derin öğrenme benzer kavramlar olmakla birlikte bazı farklılıklar bulunmaktadır. Şekil 3.1’de yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişkiler verilmiştir. Makine öğrenmesi, yapay zekânın alt dallarından biridir. Makine öğrenmesi sistemleri, sembolik yapay zekânın önemli bileşeni olan sabit kodlamanın aksine çok sayıda örnek kullanılarak oluşturulmaktadır. Çok sayıda örnek kullanılmasıyla, makine öğrenmesi sistemleri belirli kuralları öğrenerek daha önce görülmemiş veri üzerinde tatmin edici tahminlerde bulunabilmektedir. Şekil 3.2’de sabit kodlama ile makine öğrenmesi arasında farkı belirten blok diyagramları gösterilmiştir [162]. Çoğu makine öğrenmesi algoritması, satış tahminleri, öneri sistemleri, piyasa kişiselleştirme gibi problemlerle ilgili düzenlenmiş veriler üzerinde iyi performans göstermektedir. Makine öğrenmesindeki önemli faktörlerden biri olan özniteliklerin düzgün çıkarımında, araştırmacıların oldukça fazla zaman harcaması olumsuz bir durum olarak karşılanmaktadır.



Şekil 3.1. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki



Şekil 3.2. Sabit kodlama ile makine öğrenmesi arasında fark

Bilgisayar görmesi ve DDİ gibi önemli alanlarda, özellik çıkarımı yüksek boyutlar sebebiyle oldukça güç olmaktadır. Doğrusal regresyon, Karar Ağaçları, Naive Bayes, Destek Vektör Makineleri (DVM) gibi geleneksel makine öğrenmesi teknikleri; karşılaşılan benzer problemler nedeniyle yetersiz kalmaktadır. Derin öğrenmede modern teknikler ve donanımsal hesaplama gücü kullanılarak mevcut problemlerin çözülmesi hedeflenmektedir [44,163–165].

Makine öğrenmesinde kullanılan algoritmalar, verideki kümeleri temsil eden eğitim verilerini temel alan parametreleri esas alır. Eğitim verileri dış dünyayı daha gerçekçi bir şekilde temsil etmek amacıyla artırıldıkça, bu algoritmalar daha doğru sonuçlar verebilmektedir. Farklı algoritmalar, verileri farklı şekilde analiz eder. Bunlar, kullanıldıkları makine öğrenmesi tekniklerine göre kategorize edilir. Makine öğrenmesi temel olarak denetimli, denetimsiz ve pekiştirmeli öğrenme olmak üzere üç alt başlıkta gruplandırılmaktadır. Bu yaklaşımlar ile hedef kategorilerin tahmin edilmesi, olağandışı veri noktalarının bulunması ve verideki benzerlikleri bulmak için regresyon, sınıflandırma ve kümelemenin kullanılması hedeflenmektedir.

Denetimli öğrenmede, regresyon ya da sınıflandırma sürecine başlamadan önce algoritmalar belirlenen etiketli örnekler temel alınarak tahminde bulunulur. Bu yöntem, veriye ait etiket bilgisinin önceden belirlenmesi sebebiyle sonucun nasıl olabileceği öngörülen durumlarda faydalıdır. Bazı denetimli öğrenme yöntemleri aşağıda sıralanmıştır.

- o Doğrusal Regresyon
- o Yapay Sinir Ağları
- o Sade Bayes Yöntemi
- o DVM
- o En Yakın K Komşu Algoritması
- o Karar Ağaçları

Son 100 yıla göre şehirlerdeki salgın hastalıklarla ilgili bilgiler içeren bir küme sağlayıp, bir sonraki salgın sürecinin ne olacağını öğrenmek istediğimizi varsayalım. Buradaki veri kümesinde

mevcut olan nüfus, şehir, süre, ölüm oranı ve vaka sayısı gibi etiketler kullanılıp denetimli öğrenme teknikleri kullanılmaktadır.

Denetimsiz öğrenmede, kümeleme süreci öncesinde veriler etiketlenmez. Ancak algoritma, verileri düzenleyerek veya bunların yapısını açıklayarak veri noktalarını kendisi etiketleyerek kümeler oluşturulur. Bu teknik, sonucun nasıl olacağı tahmin edilemeyen durumlarda faydalıdır. Örneğin, müşteri verilerini sağlayıp benzer ürünlerden hoşlanan müşterilerin kümelerini oluşturmak istediğimizi düşünelim. Kullanılan veriler etiketlenmeden veri noktalarında keşfedilen benzerlikler temel alınarak sonuçtaki etiketler oluşturulur.

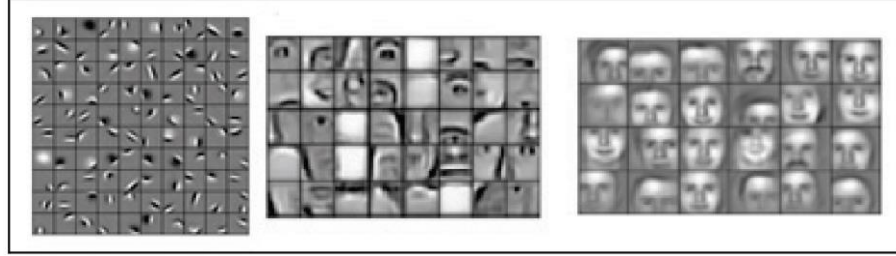
Pekiştirmeli öğrenme, sonuçlardan öğrenen ve gerçekleştirilecek eylemi kararlaştıran algoritmaları kullanır. Algoritmalar, her eylemden sonra seçeneğin doğru mu, yanlış mı yoksa yansız mı olduğunu belirleyen geri bildirimler alır. Pekiştirmeli öğrenme, insan kılavuzluğu olmadan birçok küçük kararlar alması gereken otomatikleştirilmiş sistemler için kullanılabilir bir makine öğrenmesi yaklaşımıdır. Pekiştirmeli öğrenme, psikolojideki davranışçı yaklaşımdan esinlenen, ajanın bir ortamda en yüksek ödül değerine ulaşabilmesi için hangi eylemleri yapması gerektiğine karar verebilen bir mekanizmanın oluşturulmasıyla ilgilenir. Örneğin, sürücüsüz bir araç tasarımı yaparken bu aracın yasalara uyduğu ve insan güvenliğini sağladığı net bir şekilde ortaya konulmalıdır. Aracın pekiştirmeli olarak tecrübe kazandıkça hız limitini aşmamayı, şeritte kalmayı ve yayaları görünce fren yapmayı öğrenmesi hedeflenir.

3.2. Derin Öğrenmenin Temelleri

Makine öğrenmesinin bir alt dalı olan derin öğrenme ile ilgili ilk çalışmalar 2009 yılında başlamıştır [151,157–160]. Modern teknikler kullanılarak özellik çıkartımı otomatik olarak yapılmaktadır. Örneğin, bir imgede yüz olup olmadığını tespit eden bir derin öğrenme algoritması ile bir veri seti kullanılarak ilk aşamada imgede kenar tespiti yapılır. İkinci aşamada imgede bulunan burun, gözler gibi şekiller tespit edilir. Son aşamada ise yüz şekilleri gibi karmaşık yapıya sahip olan nesneler bulunur. Her bir katman bir önceki katmandaki verinin gösterimine bağlı olarak eğitilir. Katmanlar uygun olarak bir araya getirildiğinde, yüze ait özellikler hiyerarşik bir şekilde öğrenilir [161–165].

Derin öğrenmede öne çıkan çalışmalar bilgisayar görüşü problemleri temel alınarak başlamıştır [148,166,167]. Büyük veri ve hesaplama gücünün gelişmesi ile birlikte KSA'nın kullanımı 2012 yılı itibari ile ilgi çekici hale gelmiştir [151]. En çok bilinen KSA mimarileri olarak LeNet [148], AlexNet [151], ZFNet [165], VggNet [168], GoogLeNet [169], ResNet [170] ve DenseNet [171] ön plana çıkmaktadır. KSA'lardan oluşan bu mimariler; nesne tespiti, sınıflandırma ve segmentasyon gibi bilgisayar görüşü görevlerinde yüksek performans göstermiştir. Bu modeller sonuç itibariyle birer sinir ağı olsa da klasik YSA'lara göre daha karmaşık ve veriye ait önemli ayırt edici özelliklerin otomatik olarak öğrenilmesi konusunda çok

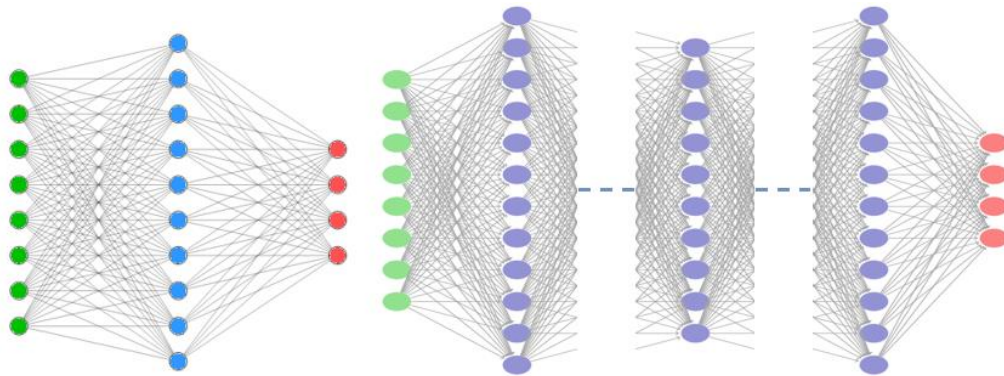
daha başarılı olmuşlardır. KSA tarafından algılanan yüz özellikleri hiyerarşik olarak öğrenilmesinde katmanlardaki görselleştirilmiş örneği Şekil 3.3'te görülmektedir.



Şekil 3.3. KSA tarafından algılanan yüz özellikleri hiyerarşisi örneği [172,173]

Derin öğrenmenin farklı alanlarda artan kullanımının sebebi, donanımsal uygunluk (Grafik İşlem Birimleri (GİB)), büyük veri, yeni algoritmalar, bulut sağlayıcılar ve yazılım araçlarındaki gelişmeler olmuştur [1,2]. Ayrıca büyük şirketlerin derin öğrenme alanına yeni başlayan kişi ve gruplarla, başlangıç aşamasında kendi problemlerinde kullanabilecekleri çok büyük veriler üzerinde kendi eğitmiş oldukları modelleri paylaşmasıdır.

Derin öğrenme kavramı, ilk olarak 1986 yılında Rina Dechter tarafından ve 2000 yılında Igor Aizenberg ve meslektaşları tarafından kullanılmıştır [183,184]. Derin öğrenmenin popüler olması ise bir KSA mimarisi olan AlexNet'in, 2012'de ImageNet yarışmasını kazanmasıyla başlamıştır [151]. DSA, Şekil 3.4'te görüldüğü gibi çok katmanlı algılayıcılardaki gizli katman sayısının artmasıyla ortaya çıkmıştır. Gizli katman sayısı ve katmanlarda kullanılan nöron sayısı arttıkça, modelin derinlik özelliği de artmaktadır. Derin öğrenmenin anlaşılması için temeli olan YSA'ların anlaşılması önem arz etmektedir. Sonraki alt bölümde YSA'lar detaylı olarak anlatılacaktır.



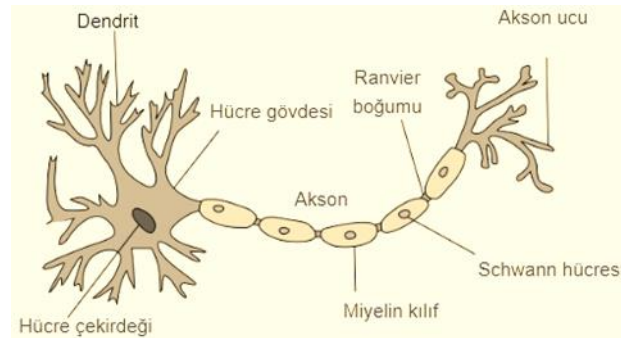
Şekil 3.4. 8 girişli ve 4 çıkışlı basit bir sinir ağı ve DSA'nın gösterimi [174]

3.3. Yapay Sinir Ağları

YSA, insan beyninin çalışma prensibinin anlaşıldığı ölçüde, merkezi sinir sisteminden esinlenerek yapılan çalışmalar ile geliştirilen, hesaplama yapabilen modellerdir. Gurney [175],

Sinir Ağlarını, girişi istenen veri tipine benzetilip etkinleştirilen bir anahtar dizisi olarak açıklar. Bu durum, nöronların beyinde nasıl çalıştığına benzemektedir. Bir başka deyişle, YSA'lar, biyolojik sinir sisteminin çalışma şekli taklit edilerek katmanlar şeklinde organize edilen birbirine bağlı nöronların bir araya gelmesiyle oluşturulan yapılardır. Basit bir YSA, nöron (sinir hücresi) dediğimiz hesaplama yapan tek bir hücreden oluşur. İnsan beyni, tahminen 100 milyar nörondan oluşur [175]. Nöronlar taşıdığı bilgiyi bağlantılı olduğu nöronlara aktararak bilgiyi işlerler. Burada Biyolojik Sinir Ağı (BSA) ile YSA'yı karşılaştırmakta fayda vardır. Bunun sebebi fonksiyonel ve yapısal olarak insana ait hayati önem taşıyan bir organ olan beynin biyolojik olarak ifade edilmesi ve insan benzeri akıllı sistemler geliştirirken bu yapının modellenmesinde, nasıl bir dönüşümün olduğunun anlaşılmasıdır.

BSA, şu ana kadar yapılan çalışmalara göre temel olarak sinaps, dentrit, akson ve hücre gövdesinden oluşmaktadır. BSA'da süreç nöronlar vasıtasıyla sağlanır. Şekil 3.5'te bir nöronun biyolojik gösterimi görülmektedir. Dentritler diğer nöronlardan gelen sinyalleri alır ve alınan bu sinyaller hücre gövdesinde toplanır. Bu sinyaller hücre gövdesinde işlendikten sonra elektriksel bir dürtü yoluyla aksonlara aktarılır ve aksonlar da bu sinyalleri Duarte ve Stahl tarafından tarif edildiği gibi diğer nöronlara iletir [176]. BSA'nın basitçe böyle bir çalışma prensibine sahip olduğundan YSA modeli benzer bir çalışma mantığı izlenerek geliştirilmiştir. BSA'daki yapı göz önünde bulundurularak benzer şekilde YSA'da nöronun matematiksel modellenmesi ilk olarak 1943 yılında Warren McCulloch ve Walter Pitts tarafından gerçekleştirilmiştir [142]. Ancak bu modelde herhangi bir öğrenme kuralı uygulanmadığı için biyolojik nöronun davranışı tam olarak elde edilememiştir. Cochocki ve Unbehauen yapmış oldukları bir çalışmada biyolojik sinir hücresinin temsilen gösterimini yaklaşık olarak Şekil 3.5'te görüldüğü gibi özetlemişlerdir [177].



Şekil 3.5. Bir nöronun-sinir hücresinin biyolojik gösterimi

Bir sonraki adımda, 1949'da Hebb, yapay bir nöronu öğrenmesi ve ardından verilen bir dizi örüntüyü tanıması için eğitmeyi mümkün kılan bir kural bulması olmuştur [178]. YSA ile ilgili yapılan ilk ciddi çalışma 1958'de Frank Rosenblatt tarafından algılayıcının (perceptron) tanıtılmasıyla olmuştur [144]. Algılayıcı Şekil 3.6'da görüldüğü gibi iki katmanlı bir ağ yapısına sahip olup basit işlemlerin yapılmasında kullanılmıştır. YSA'nın en küçük parçası olarak bilinen

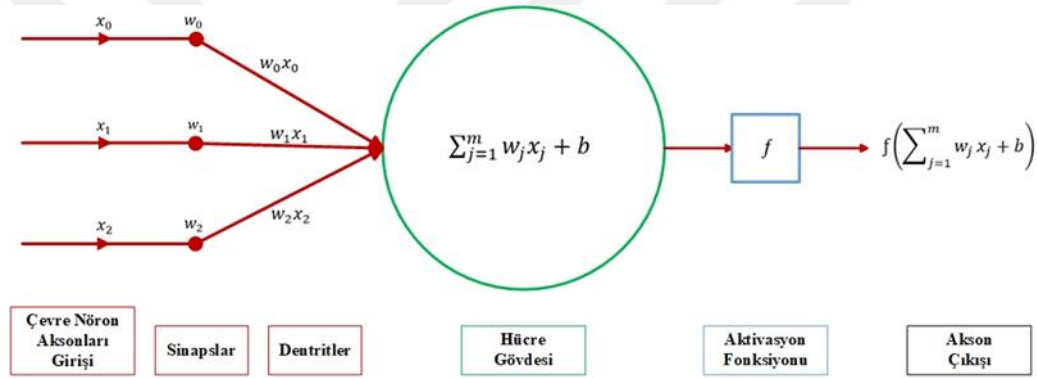
algılayıcı, Denklem 3.1'deki gibi algılayıcının matematiksel tanımı doğrusal bir fonksiyonla ifade edilmektedir. Burada w ağırlık parametresi, x bağımsız değişken, b yanlılık (bias) değeri ve y ise x 'in değerine bağlı bağımlı değişkendir. YSA'da temel amaç, en iyi performansı sağlayan w ve b parametrelerinin hesabını yapmaktır.

$$y = f \left(\left(\sum_j w_j * x_j \right) + b \right) \quad (3.1)$$

En basit sinir ağı olan algılayıcı, m sayıda x giriş vektörünün 1 ya da 0 şeklinde çıkış üretmesine dayanan bir yaklaşımdır. Bu yaklaşım matematiksel olarak aşağıdaki koşulu sağlamalıdır:

$$f(x) = \begin{cases} 1 & wx + b > 0 \\ 0 & \text{diğer} \end{cases} \quad (3.2)$$

Bir algılayıcı modelinin şematik gösterimi Şekil 3.6'da gösterilmiştir.



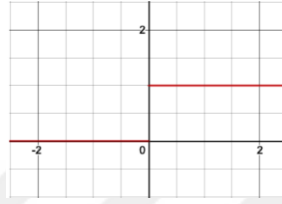
Şekil 3.6. Algılayıcı(Perceptron) matematiksel modeli

3.3.1. Çok Katmanlı Algılayıcılar

Tarihsel gelişimi itibariyle algılayıcı, tek bir doğrusal katmandan oluşur. Çoklu katmanların kullanılmasıyla birlikte çok katmanlı algılayıcı mimarisi geliştirilmiştir [179,180]. Çok katmanlı algılayıcı da giriş ve çıkış katmanları arasında bulunan katmana gizli katman denir. Giriş nöronları; dış dünyadan ağa veri girişi sağlamaktadır ve giriş katmanı olarak adlandırılmaktadır. Giriş nöronlarında herhangi bir hesaplama yapılmamaktadır. Giriş katmanından alınan veriler gizli nöronlara aktarılmaktadır. Gizli nöronların dış dünya ile doğrudan bir bağlantıları yoktur. Bu sebeple “gizli” ifadesi kullanılmaktadır. Gizli katmanlar giriş nöronlarından çıkış nöronlarına veri transfer etme ve hesaplama yapma işlevlerini yerine getirmektedirler. Çok sayıda gizli nöron gizli katmanı oluşturabilmektedir. Nöron sayısı her katman için farklı seçilebilir. Çıkış nöronları ise çıkış katmanı olarak adlandırılmaktadır ve hesaplama görevini yerine getirmektedirler ve ayrıca işlenen veriyi ağdan dış dünyaya aktarmaktadırlar.

Algılayıcının eğitilmesinde bazı problemler karşımıza çıkmaktadır. Bir nöron için en iyi ağırlık ve yanlılık seçimi yapılırken eğitim örneklerinden faydalanılır. Bilgisayar bu parametreleri

ayarlarırken nöron çıkışında hesaplanan hatanın en aza indirgenmesi hedeflenir. Somut bir örnek vermek gerekirse elimizde kedi ve köpek resimlerinden oluşan bir veri kümesinin olduğunu düşünelim. Her bir nöron imgelerdeki her bir piksel değerine karşılık gelir. Bilgisayar bu imgeleri işlerken nöronlar ağırlıklarını ve yanlılık değerlerini uygun değere getirerek hatalı tanıma işlemi minimize edilmeye çalışılır. Bu yaklaşım sezgisel bir yaklaşım olup ağırlıklarda küçük değişimler olur. Bu küçük değişim çıkışta da benzer şekilde küçük bir değişim oluşturur. Burada çıkışta büyük bir değişim varsa öğrenme gelişim göstermemektedir. Algılayıcı çıkışı Şekil 3.7’deki gibi ya 1’dir ya da 0’dır. Bu da öğrenme için büyük bir değişim olmaktadır ve öğrenmeye katkı sağlamamaktadır.



Şekil 3.7. Basit bir algılayıcı çıkışı

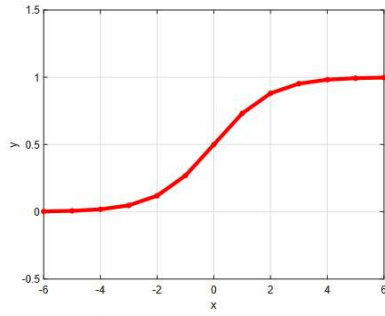
Bu olumsuzluğu ortadan kaldırıp öğrenmeye katkı sağlamak için aktivasyon fonksiyonları denilen daha düzgün bir yapıya ihtiyaç bulunmaktadır. 0’dan 1’e sürekli olarak adım adım geçiş sağlayabilen bir fonksiyon bu durumu düzeltebilir. Bu da matematiksel olarak türevini hesaplayabileceğimiz sürekli bir fonksiyona karşılık gelmektedir. Matematikte türev bir fonksiyonun bir noktadaki değişiminin ne kadar olduğunu gösterir. Burada kullanılacak aktivasyon fonksiyonlarında türev, bir grafikteki bir noktadaki teğet çizginin eğimidir. Algılayıcı özü itibariyle doğrusal bir denklem olduğu için doğrusal olmayan bir sistemin sınıflandırma problemini çözmekte yetersiz kalır. Bu nedenle, aktivasyon fonksiyonları algılayıcı için çözüm olarak geliştirilmiştir [181].

3.3.2. Yaygın Aktivasyon Fonksiyonları

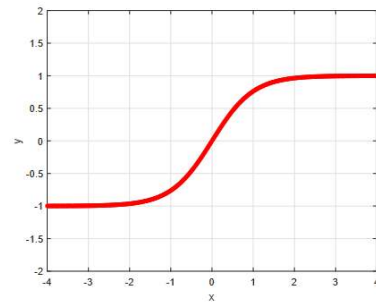
Aktivasyon fonksiyonları bir öğrenme algoritması oluşturmak için kullanılan temel yapılardan biridir. YSA çıkışına yönelik kararı, aktivasyon fonksiyonları belirlemektedir. Sigmoid, Tanh, ReLU fonksiyonları, literatürde yaygın olarak kullanılan aktivasyon fonksiyonları olarak bilinmektedir. Şekil 3.8’de bu aktivasyon fonksiyonlarına ait eğriler görülmektedir.

Sigmoid fonksiyonu Denklem 3.3’deki gibi tanımlanır. Sigmoid fonksiyonu, matematiksel olarak sürekli olup, girişin $(-\infty, \infty)$ arasında değişmesi durumunda çıkışın $(0, 1)$ arasında değiştiğini gösterir.

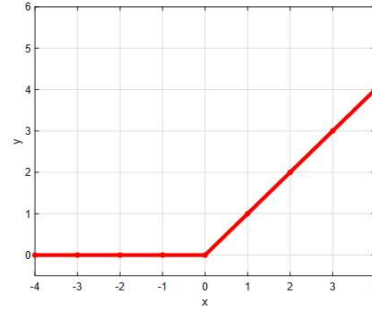
$$f(x) = \frac{1}{1+e^{-x}} \quad (3.3)$$



(a) Sigmoid fonksiyonu eğrisi



(b) Tanh fonksiyonu eğrisi



(c) ReLU fonksiyonu eğrisi

Şekil 3.8. Farklı yaygın aktivasyon fonksiyonları olan sigmoid fonksiyonu (a), tanh fonksiyonu (b) ve ReLU fonksiyonuna (c) ait grafiksel gösterimler [181]

Sigmoid aktivasyon fonksiyonlu bir nöron algılayıcıya benzer bir yapıya sahiptir. Çıkış ifadesi sürekli ve azar azar değişmektedir ve 0.5539, 0.123191 gibi değerlere sahiptir. Bu da öğrenme sürecini yavaşlatmakta ve sigmoid nöronun net sonuçlar vermesinden ziyade olabilirliği ile ilgili sonuç vermektedir. Ayrıca sigmoid çıkışının sıfır merkezli olmaması bir diğer dezavantajıdır.

Bir başka aktivasyon fonksiyonu olan tanh, Denklem 3.4'deki gibi tanımlanır ve çıkışı -1 ile 1 arasında değişen YSA'larda kullanılır.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.4)$$

Tanh aktivasyon fonksiyonu, Şekil 3.8'de görüldüğü gibi sigmoid fonksiyonuna benzer bir şekle sahiptir. Sigmoid fonksiyonuna göre avantajı ise negatif değerleri de kullanabilmesidir. Olumsuz tarafı ise sigmoid fonksiyonunda olduğu gibi üretilen çıkışlar sıfır merkezli değildir. Literatürde kaybolan eğim sorunu denilen durumla karşı karşıya kalabilmektedir.

ReLU aktivasyon fonksiyonu, Sigmoid'te karşılaşılan bazı optimizasyon problemlerini çözmek için geliştirilmiştir. Uygulamalarda çok iyi sonuçlar verdiği için popüler hale gelmiştir. ReLU doğrusal olmayan bir fonksiyon olup basitçe Denklem 3.5'teki gibi tanımlanır.

$$f(x) = \max(0, x) \quad (3.5)$$

Şekil 3.8’de görüldüğü gibi ReLU fonksiyonu negatif değerler için 0 olup, pozitif değerler için doğrusal olarak değişmektedir. ReLU fonksiyonu sigmoid fonksiyonuna göre uygulamada kullanımı donanımsal açıdan daha kolaydır.

ReLU Fonksiyonunun Bazı Yaygın Varyasyonları

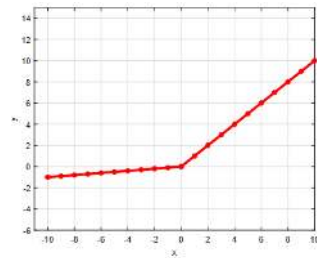
Birçok araştırmacı, ELU ve Leaky ReLU gibi ReLU fonksiyonuna dayalı bazı gelişmiş aktivasyon fonksiyonları önermiştir [182]. ELU(Exponential Linear Units-Üstel Doğrusal Birimler) x değeri 0’dan farklı olduğu değerler ve $\alpha > 0$ için aşağıdaki gibi tanımlanır. α hiperparametre olup, 0’dan küçük değerleri de hesaba katarak kaybolan gradyan etkisine çözüm getirmektedir. ReLU aktivasyon fonksiyonundan farklı olarak, ELU fonksiyonunda aktivasyonların ortalamasını 0’a yaklaştıran negatif değerler bulunmaktadır. Bu da hem ağı hızlı eğitilmesini sağlamaktadır hem de doğal gradyan inişini de takip etmektedir. ELU fonksiyonu Denklem 3.6’daki gibi belirtilen koşullar ile tanımlanır [183].

$$f(\alpha, x) = \begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (3.6)$$

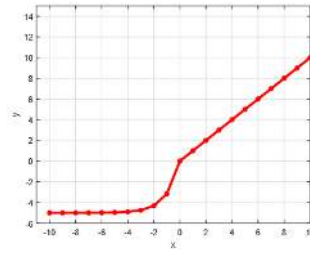
Leaky ReLU fonksiyonu, α parametresi sayesinde ReLU aktivasyon fonksiyonundaki ölü nöronlar problemini ortadan kaldırıp gradyan değerinin 0’dan farklı olmasını sağlamaktadır. α değeri Leaky ReLU için genellikle eğitime başlamadan belirlenip 0.01 olarak alınır. Leaky ReLU fonksiyonu $\alpha > 0$ için Denklem 3.7’deki gibi belirtilen koşullar ile tanımlanır [184,185].

$$f(\alpha, x) = \begin{cases} \alpha x & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \quad (3.7)$$

Şekil 3.9’da ELU ve Leaky ReLU aktivasyon fonksiyonlarına ait eğriler görülmektedir. Aktivasyon fonksiyonları, verideki örüntülerin öğrenilmesini geliştirme kabiliyetine sahip olmaları sebebiyle özellik algılama sürecini otomatikleştirir ve sinir ağlarının gizli katmanlarında kullanılır. Aktivasyon fonksiyonları derin öğrenmede gelişmeye devam etmekte olan yapılardır. Herhangi birinin diğerlerine göre üstün olduğunu söyleyememekle birlikte, derin öğrenme uygulamalarında kullanmadan önce uygun araştırma yapılmasına ihtiyaç vardır [186,187].



(a) Leaky ReLU fonksiyonu eğrisi



(b) ELU fonksiyonu eğrisi

Şekil 3.9. Yaygın ReLU aktivasyon fonksiyonu tiplerine ait grafiksel gösterimler

3.3.3. Sinir Ağlarında İleri ve Geri Yayılım

YSA'lar da ileri yayımlı ve geri yayımlı olmak üzere iki farklı mimari yaklaşım bulunmaktadır [175,176]. İleri yayımlı bir ağda bilgi sadece ileri ve tek yönde ilerlemektedir. Giriş nöronlarından bilgi alınıp gizli nöronlara ve daha sonra çıkış nöronlarına aktarılmaktadır. Gizli nöronlar yoksa şayet doğrudan çıkış nöronlarına aktarmaktadır ve çıkış katmanında çıktının üretilmesiyle süreç sonlandırılır. Dolayısıyla ağda herhangi bir kapalı döngü bulunmamaktadır. Algılayıcı ve çok katmanlı algılayıcılar ileri yayımlı ağlara örnek olarak verilebilir. Günümüzde yerini DSA'ya bırakmış olsa da bir problemin derin öğrenme yöntemleriyle ele alınması sürecinde temel başarı göstergesi olarak tercih edilmesinin bazı avantajları bulunmaktadır. Örneğin, güncel yöntemler uygulanırken asgari olarak temel bir ileri yayımlı sinir ağı modeline göre daha iyi bir performans göstermesi gerektiği fikrinden yola çıkarak, ortaya konulan yaklaşımın ne derece katkı sağladığı hakkında fikir sahibi olunabilir. Ayrıca özellik çıkarma gibi amaçlarla farklı modellerin öğrenme süreçlerine dâhil edilebilirler [180].

Çok katmanlı algılayıcılarda öğrenme, geriye yayılım ile eğitim verisi kullanılarak sağlanır. Geri yayılımın genellikle çok katmanlı sinir ağlarına özgü olduğu bilinir, ancak prensip olarak herhangi bir fonksiyonun türevini hesaplayabilir [188]. Geriye yayılım süreci algılayıcı çıkışında tespit edilen hataların adım adım düzeltilmesi olarak tanımlanmaktadır [146,189]. Yapay sinir ağlarında eğitim yapılırken, ağırlıklar ilk önce rasgele belirlenir. Daha sonra ağ, eğitim verisindeki her bir giriş için aktif hale getirilir. Giriş katmanında girişten alınan veriler gizli katmandan geçtikten sonra çıkışa doğru ileri yönde yayılarak bir tahmin yapılır. Eğitim verisindeki gerçek gözlemlenen değerler bilindiği için tahminde yapılan hatayı hesaplamak mümkündür. Ağdaki hata bir tek iterasyon ile elde edilen çıktı için hesaplanabilmektedir. Elde edilen hata değeri, maliyet fonksiyonunun eğimi ile birlikte ağı ağırlıklarının güncellenebilmesi için ağ geri yayılımla beslenmektedir. Hatayı geri yaymak için gradyan iniş gibi uygun bir optimizasyon algoritması kullanılır. Gradyan iniş kullanılmasının sebebi hatayı azaltmak amacıyla sinir ağının ağırlıklarının düzeltilmesidir. Hata önceden belirlenen bir eşik değerinin altına düşünceye kadar, girişten çıkışa doğru ileri yayılım ve hataların geriye doğru yayılım süreci belirli bir süre devam etmektedir.

YSA veya DSA ile oluşturulan model yitim-maliyet fonksiyonunun (loss function-cost fonksiyonu) minimize edilmesi için güncellenmektedir [190,191]. Maliyet fonksiyonu ile yitim fonksiyonu arasında fark, yitim fonksiyonu bir eğitim örneği için hesaplanırken maliyet fonksiyonu tüm eğitim seti için hesaplanmaktadır. YSA'da bir nöronun çıkışından ziyade her bir katmandaki ağırlıkların ayarlanması önem arz etmektedir. Bu sebeple YSA modeli, yapılacak olan tahminin doğruluğunu arttırmak için dâhili ağırlıklarını adım adım düzeltmektedir. Burada göz önünde bulundurulması gereken nokta, eğitim süreci boyunca yanlışlığı en aza indirmek için etiketlerin düzgün bir şekilde yapılması ve özniteliklerin doğru seçilmesidir.

3.4. Derin Öğrenme Yöntemleri

Bu bölümde, hiyerarşik yapıda öğrenen, tipik olarak birkaç bilgi işleme aşaması katmanını dikkate alan çeşitli DSA teknikleri incelenecektir. Ayrıca, etiketli eğitim verilerini kullanan denetimli öğrenme, etiketlenmemiş veri kümelerini analiz eden veri odaklı bir süreç olan denetimsiz öğrenme ve çevre odaklı bir yaklaşım olan pekiştirmeli öğrenme yöntemleri şeklinde çeşitli sorunları ve çeşitli öğrenme görevlerini çözmek için nasıl kullanıldıklarına bağlı olarak derin öğrenme tekniklerinin sınıflandırılması da sunulacaktır.

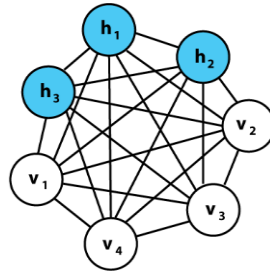
3.4.1. Denetimsiz Öğrenmede Derin Öğrenme

Denetimsiz öğrenme için yaygın olarak kullanılan derin sinir ağı teknikleri, Kısıtlı Boltzmann Makinesi (KBM), Derin İnanç Ağı (DİA), Otomatik Kodlayıcılar, Çekişmeli Üretici Ağlar (ÇÜA) ve bunların çeşitleridir [192].

Kısıtlı Boltzmann Makinesi

Amaç, verilerdeki temel yapıyı veya modeli belirlemek olduğunda, denetimsiz öğrenme yöntemleri ön plana çıkmaktadır. Popüler denetimsiz öğrenme yöntemlerinden bazıları, kümeleme, boyut azaltma, anomali algılama ve üretken modellerdir. Bu tekniklerin her biri, gizli gruplamanın belirlenmesi (latent grouping), gizli uzayın belirlenmesi, verilerdeki düzensizliklerin bulunması, yoğunluk tahmini veya verilerden yeni örnekler üretilmesi gibi farklı bir örüntü tanıma amacına sahiptir. Burada üretken modellere, özellikle Boltzmann Makinesine ve popüler varyantı KBM'ye değinilecektir.

Boltzmann Makinesi, Markov özelliğini karşılayan, olasılıksal, üretken, yönlendirilmemiş bir grafik modelidir [193,194]. Boltzmann Makinesi, aynı dağılımdan yeni örnekler oluşturmaya kadar giriş verilerinden olasılık yoğunluğunu öğrenir. Boltzmann Makinesi olasılık hesabına göre sınıflandırma yapar. Şekil 3.10'da Boltzmann Makinesine ilişkin mimari görülmektedir. Görülebilir katmandan yapılan giriş gizlenmiş katmanlara aktarılır. Gizlenmiş katmanda işlenen değerler, olumlu ya da olumsuz olsa dahi tekrardan görülebilir katmana aktarılır. Olumsuz sonuçlar tekrardan giriş değeri olarak olumlu sonuca en yakın değer olarak gizlenmiş katmanlara aktarılır.

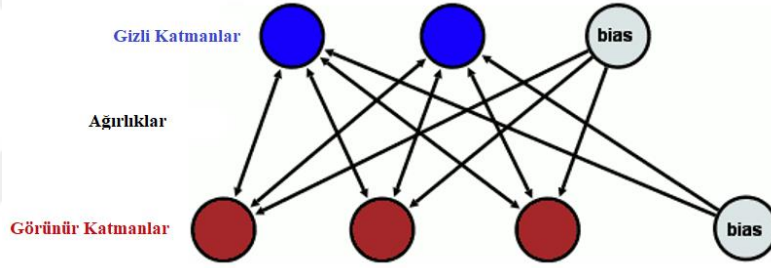


Şekil 3.10. Dört görünür katmanlı ve üç gizli katmanlı kısıtlanmamış Boltzmann Makinesi mimarisi

KBM, Ackley, Hinton ve Sejnowski tarafından 1985 yılında geliştirilen bir tür olasılıksal özyinelemeli sinir ağı (ÖSA) temeline dayanmaktadır [194]. Başka bir deyişle, KBM denetimsiz olasılıksal üretken model tipi bir sinir ağıdır [195].

Olasılıksal bir nöronda, diğer nöronlarda ve bias'tan ağırlıklı giriş ile 0 ile 1 arasında bir değer hesaplanır. Kısıtlı kelimesi, gizli katmanlar ya da görülebilir katmanlar arasında doğrudan etkileşimi engelleyen modelin iki parçalı yapısına karşılık gelmektedir. İkili vektörlerin dağılımı, verideki korelasyonun öğrenilmesi için giriş değişkenleri ile özellik çıkartıcılar arasındaki bağlantı ile modellenir.

KBM, Boltzmann Makinesinden üretilmiştir [195]. Şekil 3.11'de KBM'ye ilişkin mimari [196–198] çalışmaları esas alınarak çizilmiştir. KBM'ler derin inanç ağları oluşturma blokları olarak önerildikten sonra daha çok üzerinde durulmuştur [199]. Olasılıksal ikili nöronlara sahip bir özellik öğrenme yapısına sahiptir. KBM, görünür katman (giriş katmanı) ve gizli katman olarak iki farklı katmana sahiptir. Gizli nöronların arkasında yatan düşünce eğitim verisinden ilgili özellikleri çıkartmaktadır.



Şekil 3.11. KBM Mimarisi

KBM iki yönlü ağırlık bağlantılarına sahip i tane görünebilir nöron ve j tane gizli nörondan oluşmaktadır. Bu ağın yapısının enerjisi

$$E(v, h) = \sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{ij} h_j \quad (3.8)$$

ile tanımlanır. Burada; a_i : i görünebilir nöronuna ait bias değerini, b_j : j gizli nöronuna ait bias değerini ve w_{ij} : ikisi arasındaki ağırlığı göstermektedir. v görünebilir vektörüne ait olasılık ise

$$p(v) = \frac{1}{Z} \sum_h e^{-E(v, h)} \quad (3.9)$$

olur. Burada; Z : görünür ve gizli nöronların olası tüm durumların toplamı ile hesaplanan normalizasyon faktörüdür. Gözlenen değişkenler, görünür nöronlara giriş olarak aktarıldıktan sonra KBM'de olasılıksal bir nöron 1 değerine sahip olma olasılığına sahip olup

$$p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}) \quad (3.10)$$

olur. Burada $\sigma(x) = 1/(1 + e^{-x})$ ifadesinden faydalanılarak çözülebilir. Gizli katmanların ikili durumları kullanılarak görünür nöronların yeniden oluşturulan ikili durumları

$$p(v_i = 1|h) = \sigma(a_i + \sum_j h_j w_{ij}) \quad (3.11)$$

gibi olur. KBM'nin her bir adımı iki aşamada çalışmaktadır. İlk aşamada gizli değerler Denklem 3.10'a göre şifreleme işlemi olarak hesaplanmaktadır. Daha sonra görünür nöronlar gizli değerler kullanılarak hesaplanır. Bu işlem gözlemlenen giriş değerlerinin yeniden oluşturulmasıdır. Başka bir deyişle, kod çözme aşamasıdır. Gözlemlenen değişkenler ve yeniden oluşturulan değerler arasındaki fark, ağırlıkların güncel hesaplanmasını belirlemektedir. Bu eğitim aşaması, tek adım karşıtsal fark algoritmasıdır. Karşıtsal fark algoritması ile giriş verisinden bir dizi özellik çıkartıcılar öğrenilmektedir. Karşıtsal fark algoritması daha önce belirlenen eşik değerinin hatasına ulaşana kadar bir sonraki giriş değeri verilerek tekrar edilir. Bu nedenle ağırlık vektörü optimize edilir [200].

KBM'lerin en büyük avantajı, katmanlara ait birimler diğer katmanın bilinmesi durumunda koşullu olarak bağımsızdır. Görünen veya gizli birimler dışarıda bırakıldığında diğer katmana ait birimlerin genel olarak çarpanlarına ayrılması mümkündür. Görünen katmanın tüm birimlerinin alabileceği olası tüm değerler üzerinden integrali tek boyutlu integral şeklinde hesaplanabilir.

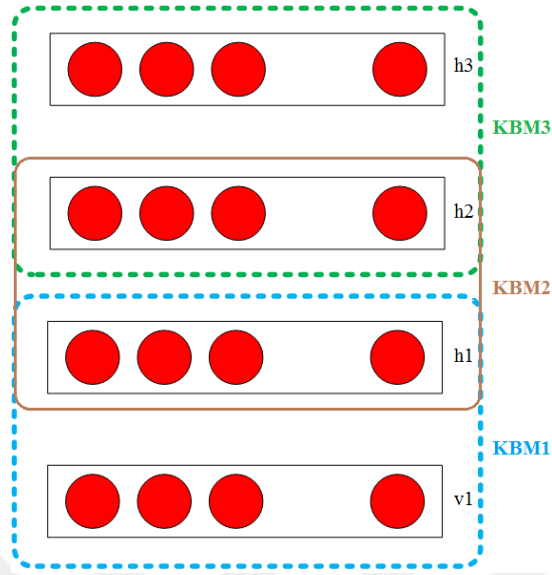
Derin öğrenmenin ilk zamanlarında, boyut azaltma ve öneri sistemleri gibi çeşitli uygulamalar oluşturmak için KBM'ler kullanılmıştır [195]. Ancak son zamanlarda, farklı makine öğrenimi uygulamalarında KBM'lerin yerini çoğunlukla Çekişmeli Üretici Ağlar (ÇÜA) veya varyasyon otomatik kodlayıcılar almıştır.

Derin İnanç Ağları

DİA'lar, gizli özellikler katmanını bir defada öğrenmek için KBM kullanılarak eğitilen olasılıksal üretken bir modeldir. 2006 yılında Geoffrey Hinton tarafından önerilmiştir [149]. İlk katmanın eğitimi tamamlandıktan sonra, KBM'nin gizli nöronlarına ait ikinci bir katman eklenir. İlk katmanın öğrenilen özellikleri giriş olarak ikinci katmana alınır ve ikinci katman aynı tarzda eğitilir. Katmanlar art arda arzu edilen katman sayısına ulaşınca kadar eklenir. Bu sebeple DİA KBM'lerin yığın haline getirilmesi ile elde edilir. Şekil 3.12'de DİA'ya ilişkin mimari [149] çalışması esas alınarak çizilmiştir. DİA'nın buna benzer stratejideki öğrenmesi denetimsiz sıralı katmansal (greedy layer-wise) öğrenme yöntemi olarak adlandırılmaktadır. Denetimsiz sıralı katmansal öğrenme, geleneksel rastgele ön değer ataması yapılan sinir ağının tersine bir ön değer atama şartı bulundurmaktadır.

Geleneksel bir DİA, giriş verisini alan görünür katman ve çoklu gizli katmandan oluşmaktadır. Her bir DİA katmanı KBM'nin eğitim yöntemine göre eğitilir. KBM1'in eğitildikten sonra, KBM2'nin gizli nöronları modele eklenir. KBM1'in gizli aktivasyonları KBM2'nin görünür katmanı olarak KBM2'ye aktarılır ve bu işlem KBM3 için de tekrar edilir. Eğitimsiz sıralı katmansal eğitimi tamamlandıktan sonra DİA modelinin ince ayarı (fine tuning) ağırlıkların optimize edilmesi için gerekmektedir. İnce ayar işlemi amaca yönelik hangi modelin

kullanılacağına bağlı olarak değişmektedir. Burada amaç üretken model ya da ayırt edici model olabilmektedir.

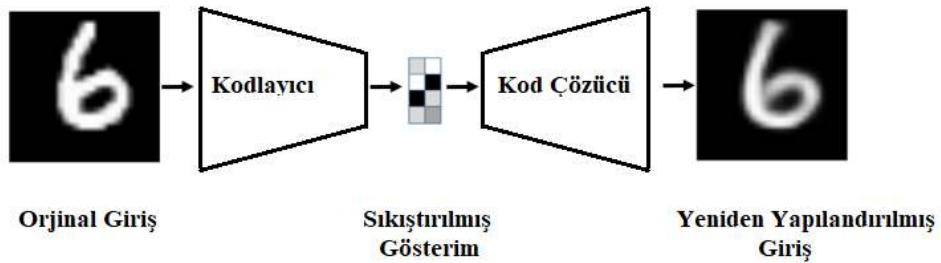


Şekil 3.12. Üç katmanlı KBM'den oluşan geleneksel bir DİA mimarisi

DİA, boyut azaltmak için denetimsiz öğrenme görevlerinde ve ayrıca sınıflandırma modelleri veya regresyon modelleri oluşturmak için denetimli öğrenme görevlerinde kullanılabilir [201,202].

Yığın/Derin Otomatik Kodlayıcılar

İleri beslemeli YSA karmaşık tanımalarda kısıtlıdır. Bundan dolayı ileri beslemeli sinir ağları fonksiyonel olarak iyileştirilerek otomatik kodlayıcı yapısı ortaya çıkarılmıştır [146,203]. Genelde bu yapıda gizlenmiş katmanlardaki nöron sayısı giriş katmanına göre daha azdır. Aynı zamanda giriş ve çıkış katmanları birbirine eşittir. Otomatik bir kodlayıcı kendi girişlerini oluşturmayı amaçlayan bir tür çok katmanlı algılayıcı gibi davranır. Şekil 3.13'te örnek bir otomatik kodlayıcı görülmektedir. Giriş görüntüsü sıkıştırılmış bir gösterime kodlanır ve ardından tekrar çözülür.

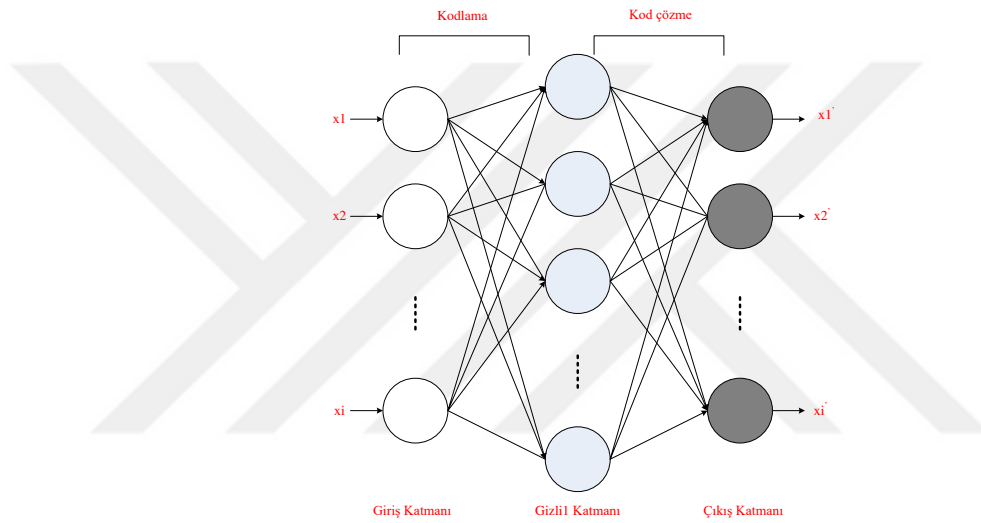


Şekil 3.13. Otomatik kodlayıcı örneği [203]

Bazı veri kümeleri, özellikler içinde karmaşık bir ilişkiye sahiptir. Bu nedenle, yalnızca bir Otomatik Kodlayıcı kullanmak yeterli değildir. Tek bir Otomatik Kodlayıcı, giriş özelliklerinin boyutsallığını azaltamayabilir. Bu nedenle, bu tür kullanım durumları için yığın otomatik

kodlayıcılar kullanılır. Yığın otomatik kodlayıcılar, adından da anlaşılacağı gibi, birbiri üzerine yığılmış birden çok kodlayıcının bir araya getirilmesiyle oluşturulur [204].

Burada model Şekil 3.14'te gösterildiği gibi çıkış hedef değerleri giriş değerleri olarak eğitilmektedir. Ağırlıklar giriş değerleri ile üretilen değerler arasındaki farka göre güncellenir. Yığın otomatik kodlayıcılarının elde edilmesi için model eğitildikten ve ağırlıklar belirlendikten sonra, çıkış katmanı çıkartılır ve ek olarak gizli katman eklenir. Daha sonra, gizli-çıkış ağırlıkları yeni gizli katmanın girişi olmaktadır. Bu aşamada, model yeni giriş değerlerinin yeniden oluşturulması için eğitilir. Bu sebeple, sıralı katmansal yaklaşımı yığın otomatik kodlayıcılarda da kullanılmaktadır.

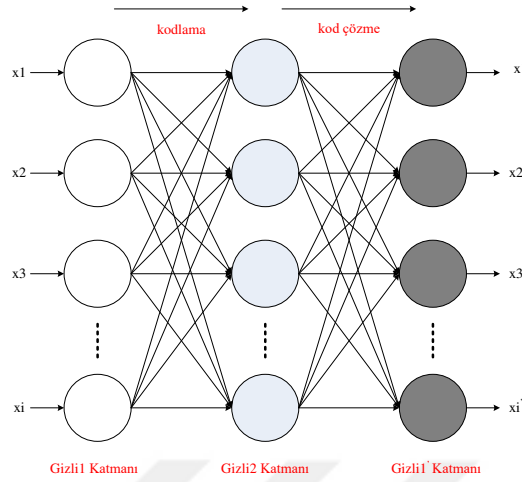


Şekil 3.14. Otomatik kodlayıcı gösterimi

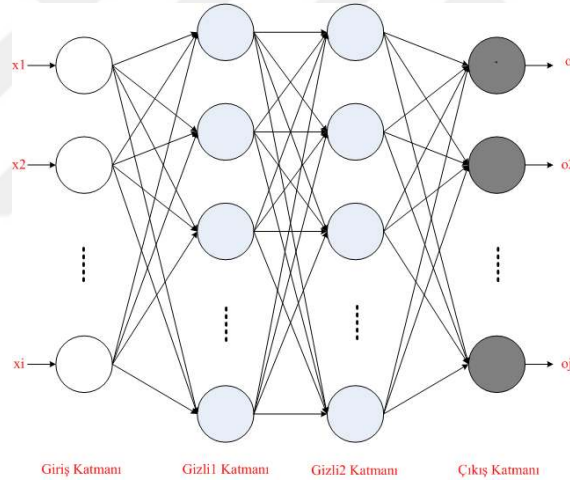
Modelin bu adımında eğitilmesinden sonra gizli1 ve gizli2 katman ağırlık değerleri belirlenir. Bu yolla gizli katmanların gerekli sayısı sıralı tarzda birbiri ardına eklenebilir. Şekil 3.15, Şekil 3.14'teki modelin bir sonraki aşamasını temsil etmektedir. Arzu edilen gizli katman sayısı sıralı katmansal eğitim tarzında eklendikten sonra çıkışın son bir katmanı Şekil 3.15'te gösterildiği gibi eklenir. Bu aşama denetlendikten sonra modelin ince ayarı için eğitim gerekmektedir. Bundan dolayı hedef sınıf değerleri geri yayılım yöntemi ile kullanılır ve bu sebeple gizli2-çıkış ağırlık değerleri belirlenir.

Otomatik Kodlayıcıların parametre güncellemesi YSA'da kullanılan yöntemler ile gerçekleştirilir. Buradaki en büyük fark çıkış katmanındaki hatanın belirlenmesinde denetimli öğrenimde kullanılan etiket bilgisi yerine giriş katmanında yer alan verinin kullanılarak hatanın hesaplanmasıdır. Belirlenen hata geri yayılım algoritması ve toplu gradyan inişi vb. optimizasyon yöntemleri kullanılarak parametre güncellemelerinin yapılmasında kullanılır [210,211]. Geleneksel otomatik kodlayıcılar boyut indirgeme için kullanılmıştır [193]. Sonraki süreçte diğer makine

öğrenmesi problemlerine de uygulanmıştır [210,212]. Şekil 3.16'da üst üste yığılmış iki kodlayıcıya sahip bir yığın otomatik kodlayıcı görülmektedir.



Şekil 3.15. Gizli katmanın giriş katmanı olarak alınması ve tekrar oluşturulması gösterimi



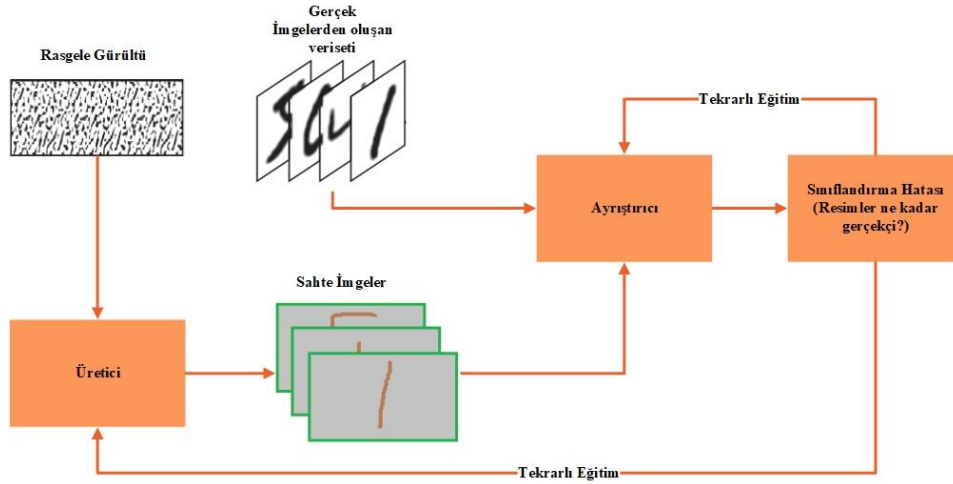
Şekil 3.16. İki katmanlı yığın otomatik kodlayıcı örneği

Çekişmeli Üretici Ağlar

Derin öğrenme tabanlı yaklaşımları kullanan üretken modelleme yöntemlerinden biri olan ÇÜA, son yıllarda oldukça popüler olmuştur [205,206]. ÇÜA, Ian Goodfellow tarafından 2014 yılında önerilmiştir [152]. Denetimsiz öğrenme problemlerini ele alan ÇÜA, üretici ağ ve ayrıştırıcı ağ olmak üzere iki ağdan oluşmaktadır. Bu iki ağ sürekli olarak birbiri ile rekabet etmektedir. Üretici ağ ile ayrıştırıcı ağ için gerçek verilerden ayırt edilemeyen sahte veriler oluşturulurken, ayrıştırıcı ağ ile gerçek veriler üretici ağ tarafından üretilen sahte verilerden ayırt edilir. Bu yapıya kalpazan ve polis ikilisi örneği verilebilir. Başlangıçta kalpazan tarafından polise sahte para gösterilir. Polis paranın sahte olduğunu tespit eder ve niçin sahte olduğunu kalpazana belirtir. Kalpazan polisten almış olduğu geri beslemeye göre yeniden sahte para yapar. Kalpazan parayı

tekrar polise gösterir. Polis paranın sahte olup olmadığını tekrar kontrol eder ve bu süreç polisin sahte parayı gerçek paradan ayırt edemeyecek duruma gelene kadar devam eder. Böylece, polis zamanla kalpazanları bulabilmek için sahte paraları yakalamada daha başarılı olurken kalpazan polisi yenebilmek için daha gerçekçi sahte paralar üretmeye başlar.

ÇÜA’da eğitilmesi gereken iki farklı YSA mimarisi vardır. ÇÜA’lar, ayrıştırıcı ağ eğitimi ve üretici ağ eğitimi birlikte yürütmelidir. Üretici ağın eğitimi ve ayrıştırıcı ağın eğitimi birbirlerinden geri bildirimler aldığından, tek bir yinelemede dönüşümlü olarak eğitim sağlanmaktadır. Üretken modeller, denetimsiz makine öğreniminin bir dalıdır. Ancak üretken modellemeye dayanan çekışmeli üretici ağı mimarisinin eğitimi, denetimli makine öğreniminin konusu olarak kabul edilir. Ayrıştırıcı ağın amacı gerçek veri dağılımından elde edilen her imge için çıkışının maksimize edilmesi ve gerçek veri dağılımından olmayan her imge için ise çıkışının minimize edilmesidir. Ayrıştırıcı ağ ve üretici ağ zıt rolleri üstlenmeleri sebebiyle çekışmeli eğitim olarak adlandırılmaktadır. Ayrıştırıcı ağ ve üretici ağ deęişen bir tarzda eğitilmektedir. Her iki yapı içinde amaç gradyan inişle optimize edilen kayıp fonksiyonu ile ifade edilmektedir. Üretici model kalpazanlık kabiliyetlerini geliştirmeye devam ederken, ayrıştırıcı model kalpazanlığı tanıma becerilerini geliştirmeyi sürdürmektedir. Ayrıştırıcı ağ, giriş imgesinin gerçek ya da üretilen bir imge olduğunu tespit ederek sınıflandırmaya çalışmaktadır. Burada önemli nokta geriye yayılımı, ayrıştırıcı ve üretici modeller üzerinden, üretici modellerin parametrelerini ayarlamak için gerçekleştirmektir. Böylece ağ bir süre sonra gerçeklerinden ayırt edilemeyen imgeleri üretmeyi öğrenmektedir. Şekil 3.17’de klasik bir ÇÜA yapısı görölmektedir. Bu yapıya bakıldığında ayrıştırıcı sahte imgeler ile gerçek imgeleri karşılaştırarak karar vermektedir. Buradaki veri seti gerçek imgelerden oluşmaktadır ve üretici kullanılarak oluşturulan imgelerin bu imgelere benzemesi beklenmektedir. Rasgele gürültüler doğrudan üreticiye verilerek sahte örneklerin oluşturulması süreci başlatılmaktadır. Üretici ağ ise ilk aşamada rasgele gürültüleri giriş olarak alıp çıkışında sahte imgeler oluşturmaktadır. Üretici ağın amacı sonraki aşamada gerçek imgelere benzer imgeler oluşturmaktır. Ayrıştırıcı ise gerçek imgeler ile sahte imgeleri karşılaştırıp üretilen imgelerin gerçek olanlara ne kadar yakın olduğunu belirlemektedir. Geriye yayılım ile ayrıştırıcının tahminlerinin ne kadar iyi olduğuna baęlı olarak ayrıştırıcı ve üretici ağlar yeniden düzenlenmektedir [207].



Şekil 3.17. Klasik bir ÇÜA yapısı [152,206]

ÇÜA'lar sayesinde makine veri üretiminin ne kadar ilerleyebildiğine dair çarpıcı bir örnek, Şekil 3.18'de gösterilen insan yüzlerinin sentezidir. 2014'te ÇÜA'lar önerildiğinde, makinelerin üretebileceği en iyi şeyin bulanık bir yüz olması bile çığır açıcı bir başarı olmuştur [157]. Şimdiye kadar olan çalışmalar, ÇÜA'lardaki ilerlemeler, bilgisayarların, kalitesi yüksek çözünürlüklü portre fotoğraflarına rakip olan sahte yüzleri sentezlemesini sağlamıştır [216–218].



Şekil 3.18. ÇÜA ile elde edilen farklı yüzler [208]

3.4.2. Denetimli Öğrenmede Derin Öğrenme

Denetimli mimariler, temel olarak Çok Katmanlı Algılayıcı (ÇKA), KSA ve ÖSA ve bunların varyantlarını içerir. ÇKA'lar 3.3.1 bölümünde ele alınmıştır. Bu bölümde KSA ve ÖSA yaklaşımları üzerinde durulacaktır [192].

Konvolüsyonel Sinir Ağları

Yann Lecun tarafından 1998 yılında önerilen KSA, imge tanıma ve sınıflandırmada oldukça etkili olması sebebiyle derin öğrenme yöntemlerinin temel tekniklerinden biri olmuştur [148]. 2012 yılında, AlexNet adlı bir KSA'nın ImageNet yarışmasında resimleri etiketlemede o zamana kadarki en iyi performansı göstermesi nedeniyle KSA popülerleşmesinde büyük bir yükseliş yaşanmıştır [151]. KSA'nın temel çalışma prensibi, görme sisteminin fizyolojisinin modern araçlarla anlaşılması ile bağlantılıdır. Özellikle Hubel ve Wiesel tarafından 1959 ve 1962 yıllarında önerilen modellerden esinlenilerek ortaya çıkmıştır [209,210]. 1980'lerde Fukushima,

Hubel ve Wiesel'in basit ve karmaşık hücreler üzerindeki çalışmalarından ilham alarak "neocognitron" modelini önermiştir [145]. Neocognitron modeli, "S-hücreleri" ve "C-hücreleri" olarak adlandırılan bileşenleri içerir. Bunlar biyolojik hücrelerden ziyade bir dizi matematiksel işlemlerden oluşmaktadır. "S-hücreleri" modelin ilk katmanında yer alır ve modelin ikinci katmanında bulunan "C-hücreleri" ile bağlantılıdır. Genel fikir, "basitten karmaşığa" kavramını yakalayıp onu görsel örüntü tanıma için bir hesaplama modeline dönüştürmektir.

KSA, özelliklerin hiyerarşik olarak öğrenilmesine olanak sağlayan bir mimariye sahiptir [148,151]. Geleneksel görüntü işlemede, nesne tanıma problemlerinde görüntüdeki kenarlar gibi özelliklerin tespit edilmesine yönelik manuel teknikler uygulanırken, KSA modelinde bu özellikler otomatik olarak keşfedilir. KSA'lar, klasik çok katmanlı sinir ağlarından farklı olarak bir dizi konvolüsyon ve örnekleme katmanı (pooling layer) içermektedir. Örnekleme ile katmanların daha önemli özelliklerinin tutulması ve sonraki katmanlara boyutun düşürülerek aktarılması (subsampling) sağlanır. Filtre (kernel) boyutu, filtre sayısı, aktivasyon fonksiyonu ve örnekleme katmanı tercihi KSA modelinin başarısını etkileyen önemli parametrelerdir [211].

KSA; özel bir YSA türü olup veride boyut indirgeme avantajına sahiptir. Modeldeki derinlik fonksiyonların bir dizi dönüşümleri oluşturması anlamına gelmektedir. Bu dönüşümler girişleri orta düzey özelliklere ve çıkışlara dönüştürmek için yapılmaktadır. İmge sınıflandırma, imgenin her bir giriş pikselini tüm giriş sinir birimlerine atamayı gerektiren geleneksel bir sinir ağı ile yapılmaktadır. Bu şu anlama gelmektedir: Veri olarak ağırlıkların çok sayıda bağlantı ve hesaplaması ağ üzerinde geri ve ileri yayılmaktadır. KSA'da giriş katmandaki her bir nöron, imgedeki bölgeyi filtrenin boyutuna göre almaktadır. Nöronların ağ benzeri yerleşimi bir sonraki katmanın giriş imgesini temsil etmektedir ve bu ağ stratejisi ağ tasarım gereksinimlerine göre sürmektedir.

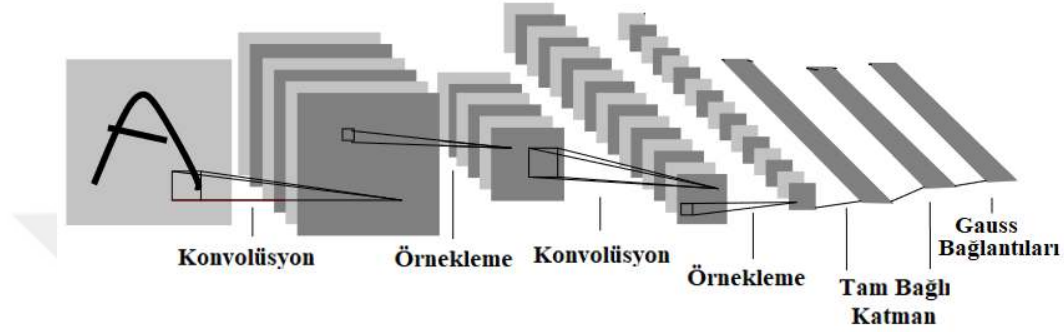
İmgeler KSA modeline giriş olarak verildikten sonra önceden belirlenmiş ya da öğrenilmiş katsayılarla sahip farklı filtreler ile konvüle edilir. $n \times n$ boyutunda bir w filtresinin kullanıldığını düşünelim: konvolüsyon sonucu (x) , $l-1$ katmanındaki her bir y imge pikselini uygun filtre katsayısı ile çarparak hesaplanmaktadır. Sonuçlar Denklem 3.12'ye göre hesaplanmaktadır. İmge boyutu, satır ve sütunların sayısından $n-1$ 'e düşürülmektedir.

$$x_{i,j}^l = \sum_a^n \sum_b^n w_{ab} y_{(i+a)(j+b)}^{l-1} \quad (3.12)$$

Doğrusalsızlık dönüşümü Denklem 3.13'te verilen konvolüsyon ile üretilen her bir değere uygulanmaktadır.

$$y_{i,j}^l = \frac{1}{1 + e^{-x_{i,j}^l}} \quad (3.13)$$

Her bir filtre ile özellik haritaları (feature maps) gibi giriş imgesinin farklı özellikleri ortaya çıkmaktadır. Konvolüsyonun çıkış imgeleri örnekleme katmanı ile alınır ve alt-örnekleme gerçekleşir. Bu adımdaki işlem boyut indirgeme ve önemli bilgilerin korunması için yapılır. Bazı alt-örnekleme işlemleri kullanılmasına rağmen en çok kullanılanlar 2x2 ortalama ve maksimum örnekleme olanlardır. 2x2 ortalama örnekleme, bitişikteki dört piksel değerinin ortalama değerinin hesaplanması ve maksimum örnekleme ise bitişik dört pikseldeki en yüksek değerli olanın alınması anlamına gelmektedir. Şekil 3.19’da KSA içerisindeki yayılıma ilişkin LeNet örneği verilmiştir.



Şekil 3.19. Konvolüsyonel ve örnekleme katmanlarından oluşturulmuş LeNet mimarisi [148]

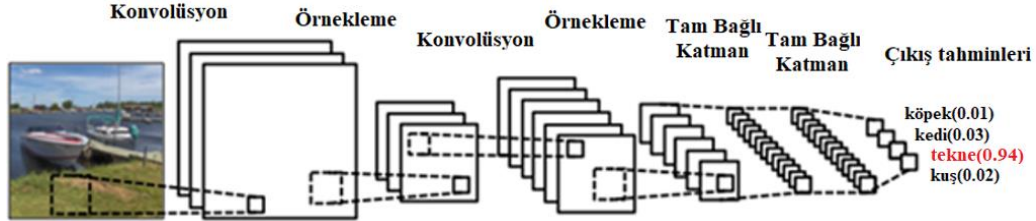
KSA’da katmanlar ardı ardına kullanılabilir. Son örnekleme işleminden sonra geleneksel YSA çalışmasına devam edilmektedir. Sonuç özellik haritası YSA ile kategorize edilmektedir. Konvolüsyonel ve alt-örnekleme katmanlarının son çıkışı tam bağlı katmana (gizli katman) aktarılır. Birden fazla tam bağlı katman kullanılabilir ve son olarak softmax fonksiyonu çıkış katmanında son karar için kullanılır. Eğitim denetimli tarzda elde edilir ve geleneksel geri yayımlı algoritma tüm modeli eğitmek için kullanılır.

Burada KSA’nın imgeler üzerindeki başarısının anlaşılması için LeNet mimarisinin çalışması incelenecektir. LeNet mimarisi ilk KSA’lardan biri olup derin öğrenme alanının gelişimine katkıda bulunmuştur. Yann LeCun tarafından yapılan bu öncü çalışma 1998’den beri başarılı tekrarları yapıldıktan sonra LeNet5 olarak adlandırılmıştır. O zamanlar LeNet mimarisi sayıların okunması gibi karakter tanımda esas olarak kullanılmıştır [148,212].

LeNet’in daha gelişmiş versiyonları son yıllarda önerilen yeni mimariler olmuştur. Bu mimarilerin çoğu LeNet’ten yola çıkarak oluşturulmuştur ve bir önceki mimarinin anlaşılması durumunda anlaşılabilirliği daha kolaydır. LeNet mimarisi dışındaki diğer mimariler incelenecek olursa, 1990’lı yıllardan 2012’ye geçen süre zarfında KSA alanı oldukça durgun geçmiştir. Veri ve hesaplama gücünün gelişmesi ile birlikte KSA’nın kullanımı 2012 yılı itibarı ile ilgi çekici hale gelmiştir [151]. En çok bilinen KSA mimarileri LeNet [148], AlexNet [151], ZFNet [165], VggNet [168], GoogLeNet [169], ResNet [170] ve DenseNet [171] mimarileridir.

Şekil 3.20’deki KSA, LeNet orijinal çalışmasındakine benzer olup giriş imgesini köpek, kedi, tekne ve kuş gibi dört farklı kategoriye sınıflandırmaktadır. Giriş verisi olarak tekne imgesi

alındığında ağ tekneyi dört ayrı kategoride en yüksek olasılıkla (0.94) atamıştır. Çıkış katmanındaki olasılık toplamı 1 olmalıdır. Şekil 3.20’de KSA her aşamada farklı işlemler yapmaktadır. Sırayla konvolüsyon, DDB, örnekleme ve sınıflandırma (tam bağlı katman-fully connected layer) aşamalarından oluşmaktadır. Bu aşamalar tüm KSA’larda geçerlidir. Dolayısıyla bu aşamalarda ne tür işlemlerin yapıldığını anlamak KSA’nın anlaşılmasında önem arz etmektedir.

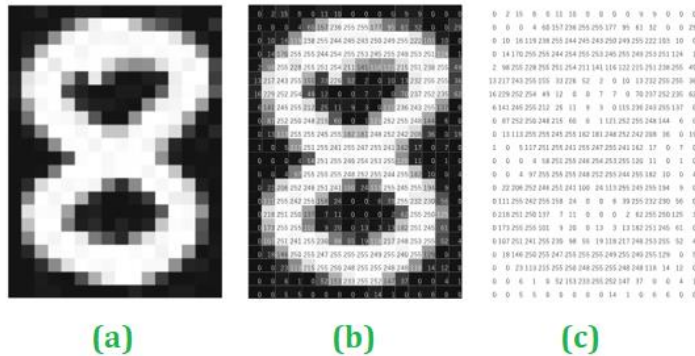


Şekil 3.20. Basit bir KSA yapısı örneği [148]

KSA kavramı *konvolüsyon* işleminden türetilmiştir. KSA’larda konvolüsyonun en temel amacı giriş imgesine ait özelliklerin çıkartılmasıdır. Konvolüsyon, giriş verisinin küçük karelerini kullanarak imgenin özelliklerini öğrenme ile piksel değerleri arasındaki uzaysal ilişkiyi korumaktadır. Burada KSA’nın imgeler üzerinde nasıl çalıştığı anlaşılmaya çalışılacaktır. KSA, konvolüsyon işlemi için

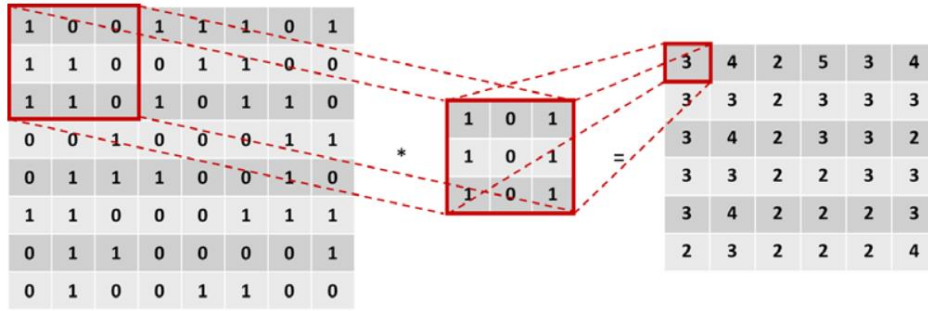
$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k] \quad (3.14)$$

kullanılır. Her imge piksel değerlerinin matrisel formu ile temsil edilebilmektedir. Kanal, bir imgenin belirli bir bileşenine karşılık kullanılan geleneksel bir terimdir. Dijital kameradan elde edilen bir imge kırmızı, yeşil ve mavi olmak üzere üç farklı kanaldan oluşmaktadır. İmge, her biri 0 ile 255 arasında piksel değerlerine sahip üç tane iki boyutlu matrisin yığın şeklinde bir araya getirilmesi olarak düşünülebilir. Gri ton bir imge ise sadece bir kanaldan oluşmaktadır. Her bir kanal iki boyutlu matrisel formda olup her bir piksel değeri 0 ile 255 arasında değişmektedir. 0 siyaha 255 ise beyaza karşılık gelmektedir. Bu bölümde bahsedilen imgeler gri tonda düşünülmelidir. Şekil 3.21 (a)’daki imgenin gri tonlarda piksel değerleri Şekil 3.21 (b, c)’de verilmiştir.



Şekil 3.21. (a) 8 imgesi ve (b, c) piksel değerleri ile temsili [213]

8x8 boyutunda her bir piksel değeri 0 ya da 1 olan bir imgeyi 3x3 boyutunda bir filtre ile konvolüsyonu hesaplanacak olursa sonuç Şekil 3.22'deki gibi elde edilecektir.



Şekil 3.22. İmge, kernel ve özellik haritasının kullanıldığı konvolüsyonel süreç

Şekil 3.22'de orijinal imge üzerinde sürekli olarak soldan sağa doğru bir filtre kaydırılarak elementer çarpımı hesaplanır ve çarpım çıktıları çıkış matrisinin tek bir elemanını oluşturan son tamsayıyı belirlemek için birbirine eklenir. 3x3 boyutlu filtre her bir adımda giriş imgesinin bir kısmını hesaba katmaktadır. KSA terminolojisinde 3x3 boyutlu matris filtre, kernel ya da özellik bulucu olarak adlandırılmaktadır. Filtrenin imge üzerinde kaydırılarak oluşturulan matris ise aktivasyon haritası (activation map) ya da konvüle edilen özellik (convolved feature) ya da özellik haritası (feature map) olarak adlandırılmaktadır. Şekil 3.22'den anlaşıldığı üzere farklı filtre matrisi değerleri aynı giriş imgesi için farklı özellik haritaları üreteceği aşikardır.

Pratik olarak KSA, bu filtrelerdeki değerleri kendi başına eğitim aşamasında öğrenmektedir. Eğitim aşamasından önce filtre sayısı, filtre boyutu ve ağ mimarisi gibi parametrelerin belirlenmesi gerekmektedir. Kullanılan filtre sayısı arttıkça daha çok imge özelliği çıkartılmaktadır ve ağ görülmeyen imgelerdeki örüntüyü tanımada daha iyi olmaktadır.

Özellik haritasının boyutu, konvolüsyon işlemine başlamadan belirlenmesi gereken üç parametre ile kontrol edilmektedir. Bunlar derinlik (depth), adım uzunluğu (stride) ve sıfır doldurma (zero-padding) parametreleridir [211]. ReLU, Bölüm 3.3.2'de anlatıldığı gibi nonlineer bir işlem olup, her konvolüsyon işleminin akabinde uygulanır. Derinlik, konvolüsyon işlemi için kullanılan filtre sayısını belirtmektedir. Şekil 3.20'de gösterilen ağda üç farklı filtre kullanılarak tekne imgesine ilişkin Şekil 3.23'teki gibi konvolüsyon işlemi uygulanmaktadır. Böylece üç farklı özellik haritası oluşturulmuştur. Dolayısıyla derinlik sayısı üç olmaktadır. Adım uzunluğu, filtre matrisinin giriş matrisi üzerinde kaydırıldığı piksel sayısına karşılık gelmektedir. Adım uzunluğu 1 ise her seferinde filtreler 1 piksel kaydırılmaktadır. 2 olduğunda ise aynı şekilde 2 piksel kaydırılarak işlem yapılmaktadır. Adım uzunluğu arttıkça daha küçük boyutta özellik haritaları oluşmaktadır.

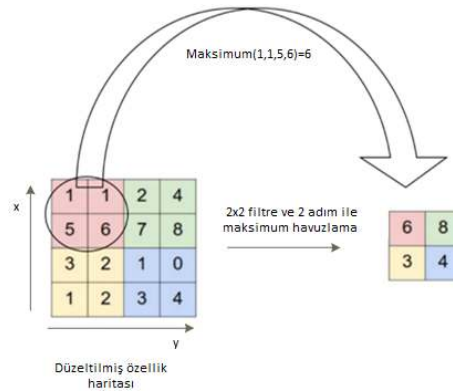


Şekil 3.23. 3 farklı derinlikli özellik haritası

Sıfır doldurma, giriş matrislerini sınırlarda sıfır ile doldurmak uygun olmaktadır. Böylece filtre giriş imgesi matrisine uygulanabilmektedir. Sıfır doldurmanın iyi yanı özellik haritası boyutunu kontrol edebilme kolaylığı sağlamaktadır. Sıfır doldurma ekleme geniş konvolüsyon olarak adlandırılmaktadır.

Örnekleme ile her bir özellik haritasının boyutu azaltılır ve en önemli bilgiyi tutmaya devam edilir. Örnekleme maksimum (max pooling), ortalama (avg pooling) ve toplam (sum pooling) gibi farklı tiplerde olabilmektedir. Maksimum örnekleme olayında uzaysal bir komşuluk tanımlanır ve bu komşuluk içerisinde düzeltilmiş özellik haritasından en büyük eleman alınır. En büyük eleman yerine ortalama ya da toplam eleman değeri de alınabilir. Ancak pratikte çoğu çalışmada maksimum örneklemenin daha iyi çalıştığı görülmüştür.

2x2 boyutlu pencere iki adım uzunluğu ile kaydırılmıştır ve her bölgede maksimum değer alınmıştır. Şekil 3.24'te görüldüğü gibi bu işlem özellik haritasının boyutunu azaltmaktadır.



Şekil 3.24. Maksimum örnekleme örneği

Örnekleme işlemi özetle giriş gösterimlerini/özelliklerini daha küçük ve kontrol edilebilir yapmaktadır. Ağdaki hesaplamaların ve parametrelerin sayısını azaltmaktadır. Böylece aşırı uyma (overfitting) kontrol edilebilmektedir. Ağ küçük değişim, bozulma ve dönüşümlerden etkilenmemesini sağlamaktadır. Giriş imgesinin ölçekten bağımsız görünümünün elde edilmesini kolaylaştırmaktadır.

Tam Bağlı Katman (TBK), çıkış katmanında softmax aktivasyon fonksiyonunu kullanan geleneksel çok katmanlı algılayıcıdır. Tam bağlı ifadesi önceki katmandaki tüm nöronlar sonraki

katmandaki tüm nöronlara bağlı olduğunu belirtmektedir. Konvolüsyon ve örnekleme katmanları çıkışı, giriş imgesinin yüksek seviyeli özelliklerini temsil etmektedir. TBK'nın amacı bu özelliklerin giriş imgesini eğitim veri kümesine bağlı olarak farklı sınıflara ayırmak için kullanmaktır. Örneğin, uygulamak istenilen imge sınıflandırma uygulaması Şekil 3.20'de gösterildiği gibi dört olası çıkışa sahiptir. TBK ekleme sınıflandırma dışında bu özelliklerin nonlinear kombinasyonlarının öğrenilmesinin basit bir yoludur. Konvolüsyonel ve örnekleme katmanlarından çoğu özellikler, sınıflandırma uygulaması için uygun olmaktadır. Fakat bu özelliklerin kombinasyonları daha iyi olmaktadır.

Son TBK'ya uygulanan aktivasyon fonksiyonu genellikle diğerlerinden farklıdır. Her göreve göre uygun bir aktivasyon fonksiyonu seçilmelidir. Çok sınıflı sınıflandırma görevine uygulanan bir aktivasyon fonksiyonu, her bir değerin 0 ile 1 arasında değiştiği ve tüm değerlerin toplamının 1 olduğu, son TBK'dan çıktı gerçek değerlerini hedef sınıf olasılıklarına normalleştiren bir softmax fonksiyonudur. Son TBK'da kullanılan aktivasyon fonksiyonunun tipik seçimleri farklı görev türleri için değişmektedir. Sigmoid fonksiyonu ikili sınıflandırma ve softmax fonksiyonu çoklu sınıflandırma için kullanılır [213].

Bu alt bölümde anlatılanlar özetlenirse; konvolüsyon ve örnekleme katmanları özellik çıkartmada kullanılmıştır. TBK ise sınıflandırıcı olarak kullanılmıştır. Şekil 3.20'ye bakıldığında giriş imgesinin tekne olması sebebi ile hedef olasılık tekne için 1, diğer üç sınıf için ise 0 olmaktadır.

Konvolüsyonel sinir ağının eğitimi aşağıdaki gibi özetlenebilir:

Aşama 1: Tüm filtreler ve parametreler ya da ağırlıklar rastgele değerler ile atanır.

Aşama 2: Ağ eğitim imgesini giriş olarak aldıktan sonra, ileri yayılım aşaması gerçekleşmektedir (konvolüsyon, DDB ve örnekleme işlemleri sonrasında TBK'da ileri yayılım) ve her bir sınıf için çıkış olasılıkları bulunmaktadır.

Aşama 3: Çıkış katmanındaki toplam hata değeri hesaplanır:

$$\text{Toplam Hata} = \sum \frac{1}{2} (\text{hedef olasılığı} - \text{çıkış olasılığı})^2 \quad (3.15)$$

Aşama 4: Geri-yayılım tekniği, ağdaki tüm ağırlıklar ile ilgili hata eğimlerini hesaplamak için kullanılır. Ardından eğim inişi, çıkış hatasını minimize etmek amacıyla filtre değerlerinin ya da ağırlık ve parametre değerlerinin güncellenmesi için kullanılır.

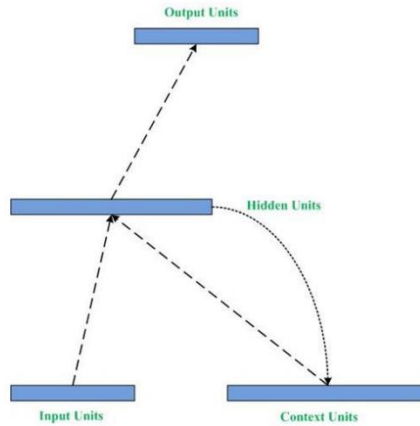
Aşama 5: Eğitim kümesindeki tüm imgeler ile aşama 2-aşama 4 arası yakınsama gerçekleşinceye kadar tekrar edilir.

Yukarıdaki 5 aşama KSA'nın eğitilmesi aşamasında kullanılmaktadır [151,157,213]. Burada KSA'nın tüm ağırlıkları ve parametreleri eğitim kümesindeki imgeleri doğru olarak sınıflandırmak için optimize edilmektedir. Yeni bir imge ağa giriş olarak verildiğinde ağ ileri yayılım yönünde çalışacak ve her bir sınıf için olasılık değeri üretecektir. Eğitim kümesi yeterince büyük ise ağ yeni imgeleri iyi bir şekilde genelleştirmekte ve bu imgeleri doğru kategorilere ayırabilmektedir.

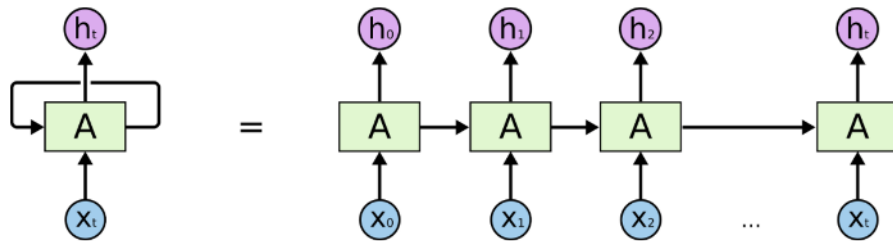
Özyinelemeli Sinir Ağları

ÖSA, sıralı veri işleme problemlerinde kullanılan, hatırlayabilen ve girdilerin birbiri ile ilişkili olduğu bir derin öğrenme yöntemidir [146]. Hesaplanmış önceki verileri belleğinde tutarak bilgi akışının sürekliliğini sağlayan bir yapıya sahip olması sebebiyle geleneksel ileri beslemeli sinir ağlarına göre daha etkilidir. Özellikle DDİ alanında sıralı veri işleme özelliği olmasından dolayı sıklıkla tercih edilmektedir [214]. Konuşma, metin, resimlere altyazı koyma gibi sıralı akışa sahip verilerde başarılı sonuçlar üreten ÖSA yapısında, klasik sinir ağı yapısından farklı olarak ilerideki katmanlardan girişlere geri besleme yapılmaktadır [215,216]. Ağ bu sayede önceki bilgilere bağlı bir çıkış oluşturmaktadır.

Elman tarafından önerilen en basit ÖSA yapısı Şekil 3.25'te görülmektedir [147]. Bu yapının daha geniş bir şekilde ifade edilmesi ise Şekil 3.26'da görülmektedir. Burada ÖSA yapısına bakıldığında aynı sinir ağın çoklu sayıda çoğaltılması gibi düşünülebilmektedir ve her bir ağın bilgiyi sonra gelen ağı aktarmaktadır. Bu bağlantılı yapı sıralı veriler ve listelerde kullanımında oldukça makul olduğu görülmektedir. ÖSA modelinin faydalı yanına bakıldığında bir önceki veri parçası ile ilişki kurar ve bu sayede girdiler unutulmadan ilerleme sağlanır.



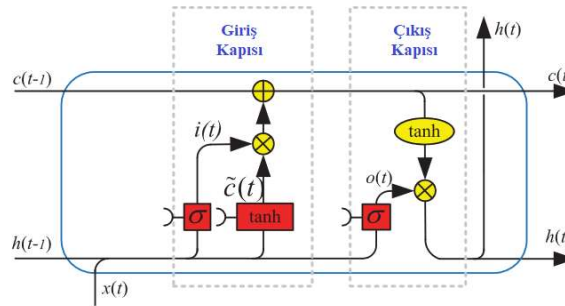
Şekil 3.25. Basit bir ÖSA örneği [147]



Şekil 3.26. Bir ÖSA'nın ileriye dönük hesaplamasında yer alan hesaplama zamanında açılması

Teorik olarak ÖSA'lar artarda gelen uzun sıralı bilgileri kullanabilmektedir. Ancak, pratik olarak sadece birkaç adım geriye gidebilirler. Bununla birlikte, özyinelemeli ağlarda önemli eksiklikler bulunmaktadır. ÖSA bir cümledeki kelimeleri hafızasında tutarken başka bir cümleden

anlamsal çıkarımda bulunup önceden öğrendiği kelimeyi yerinde kullanması gerekirken başarısız olmaktadır. Örnek verecek olursak; “Ben Türkiye’de yaşıyorum.” bilgisinden sonra gelebilen birçok cümleden dolayı “Akıcı olarak ... konuşurum.” cümlesinde ki boşluğa doğrudan Türkçeyi eklemek mümkün olmamaktadır. Çünkü cümle akışı içerisinde farklı diller yer almış olabilir ve dolayısıyla model hangisini bu boşlukta kullanacağına karar verememektedir [217]. Genel olarak ÖSA yapısının olumsuz görünen tarafı geriye yayılımda kaybolan (vanishing) gradyan ve patlayan (exploding) gradyan diye bilinen problemlerin ortaya çıkmasıdır. Gradyan tüm ağırlıkları ayarlamamızı sağlayan bir değerdir. Ancak birbirine bağlı uzun ağlarda hatanın etkisi oldukça düşerek gradyan kaybolmaya başlayabilir. Bu da öğrenmenin gerçekleştirilmesine pek katkı sağlamamaktadır. Bir başka ifadeyle geriye yayılım ile birlikte ağırlıkların yok olması mimarisinden dolayı ÖSA için kaçınılmaz bir problemidir. Her bir katmandaki ağırlıklar zincir kuralı üzerinden ayarlandığından, gradyan değerleri geriye doğru adım ilerledikçe katlanarak küçüldükten sonra sonunda yok olabilmektedir. Bu duruma çözüm olarak Hochreiter ve Schmidhuber 1997 yılında, Şekil 3.27’de görülen uzun kısa vadeli bellek ağları (UKSBA, Long Short Term Memory-LSTM) önermiştir [218]. Daha sonraki yıllarda birçok araştırmacının katkısıyla gelişen ve popülerleşen UKSBA, şu anda sıralı verilerin yapısının anlaşılmasında geniş bir kullanım alanına sahiptir.



Şekil 3.27. Orijinal UKSBA mimarisi [214]

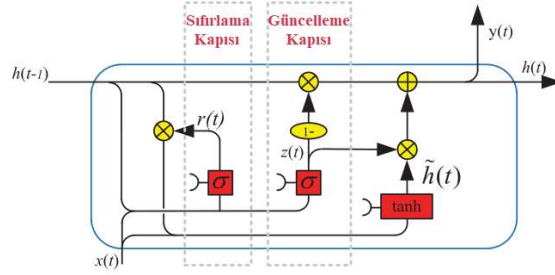
ÖSA'lara benzer yapıda olmalarıyla birlikte UKSBA hücrelerindeki işlemler açısından farklılık göstermektedir [219]. Hücrelerinde gerçekleşen bazı işlemlerden dolayı UKSBA'lar bilgiyi unutabilir ya da hafızada tutabilirler. UKSBA'larda bulunan temel farklılık yapılarında bulunan hücre durumu ve kapılardan kaynaklanmaktadır. Hücre durumu, sıralı bir dizi halinde ilgili veriyi bir tür taşıma görevi görür. Ağın hafıza birimi olarak düşünülebilir. Teorik olarak hücre durumu ilgili bilgiyi tüm dizinin işlenmesi süresince taşımaktadır. Hatta zamansal açıdan kısa vadeli bellek etkisini azaltarak daha önceki aşamalardaki bilgiler ile daha sonraki bilgiler arasında bir bağlantı kurulmaktadır. Hücre durumu bu süreçte devam ettikçe bilgi hücre durumuna kapılar vasıtasıyla ya çıkartılır ya da eklenir. Kapılar hücre durumunda hangi bilgilerin müsaade edileceğine karar veren farklı yapıya sahip sinir ağlarıdır. Kapılar eğitim aşamasında hangi bilgilerin tutulacağı hangi bilgilerin unutulacağını amaca uygun olduğunu öğrenirler. Kapılar

sigmoid aktivasyon fonksiyonlarını içerirler. Sigmoid aktivasyon fonksiyonları çıkışında 0 ile 1 arasında değerler üretmektedir. Bu da sigmoid fonksiyon çıkışının 0 olması durumunda değerlerin unutulması, 1 olması durumunda ise değerlerin aynı kalması sağlanır. Böylelikle ağ hangi değerlerin önemli hangi değerlerin önemsiz olduğunu öğrenebilmektedir.

Bir UKSBA hücresinde bilgi akışını düzenleyen unutma, giriş ve çıkış olmak üzere üç farklı kapı öğrenme sürecine dâhil edilir. Bu şekilde, modelin geçmişe yönelik hafızasının hangi şartlarda ve hangi oranda sonuca etki edeceğine karar veren bir sistem elde edilir. Unutma kapısı, ilk kapı olup hangi bilgilerin tutulması gerektiğine karar verir. Bir önceki gizli durumdan alınan bilgiler ve o anki giriş katmanından alınan bilgiler sigmoid fonksiyonundan geçirilir. Sigmoid fonksiyonu çıkışı 0 olunca bilgi unutulur, 1 olunca bilgi tutulur. İkinci kapı olan giriş kapıları da hücre durumlarının güncellenmesi için kullanılır. İlk olarak bir önceki gizli durum ve giriş katmanı sigmoid fonksiyonundan geçirilir. Bu da değerlerin 0 ile 1 arasına dönüştürülerek hangi değerlerin güncelleneceğine karar vermektedir. Ayrıca ağın düzenlenmesi amacıyla gizli durum ve o anki giriş tanh fonksiyonundan geçirilip değerler -1 ile 1 arasına sığdırılır. Daha sonra tanh çıkışı ile sigmoid çıkışı çarpma işlemine maruz bırakılır. Böylece sigmoid çıkışı tanh çıkışındaki hangi bilgilerin önemli olduğuna karar verebilmektedir. Bu iki kapıdaki işlemler tamamlandıktan sonra hücre durumunun hesaplanması için yeterli miktarda bilgi elde edilir. Burada hücre durumu ilk olarak unutma vektörü ile noktasal olarak çarpılır. Bu da 0'a yakın değerlerle çarpılması halinde hücre durumundaki değerlerin düşme ihtimalini ortaya çıkarabilmektedir. Daha sonra giriş kapısından çıkışlar alınır ve hücre durumunu sinir ağına uygun bulunduğu yeni değerlerle güncelleyen noktasal bir toplama işlemi yapılır. Bu da yeni hücre durumunu verir. Son kapı olan çıkış kapısı bir sonraki gizli durumun ne olması gerektiğine karar vermektedir. Gizli durum önceki girişlerdeki bilgileri içerir ve tahminde bulunmak için kullanılır. Burada ilk önce bir önceki gizli durum ve o anki giriş sigmoid fonksiyonundan geçirilir. Ardından güncellenen bu hücre durumu tanh fonksiyonuyla işleme tabi tutulur. Tanh çıkışı ile sigmoid çıkışı çarpılarak çıkış olan gizli durumun hangi bilgiyi taşıması gerektiğine karar verilir. Daha sonra yeni hücre durumu ve gizli durum bir sonraki adıma aktarılır. Özetleyecek olursak unutma kapısı hangi bir önceki adımlardan gelen hangi bilgilerin uygun olduğuna karar verir. Giriş kapısı ise bulunduğu aşamada hangi bilginin eklenmesi gerektiğini belirler. Çıkış kapısı da bir sonraki gizli ne olması gerektiğine karar verir.

Yeni nesil bir ÖSA olan Kapı Özyinelemeli Birim (KÖB-GRU), UKSBA'ya benzer bir yapıya sahiptir. KÖB'ler hücre durumundan kurtarılmıştır ve bilgi aktarmak için gizli durumu kullanırlar. KÖB, sıfırlama kapısı ve güncelleme kapısı olmak üzere sadece iki kapıya sahiptir. Güncelleme kapısının UKSBA'nın unutma ve giriş kapılarına benzer bir rolü bulunmaktadır. Hangi bilginin atılması gerektiğine ve hangi yeni bilginin eklenmesi gerektiğini belirler. Sıfırlama kapısı ise geçmiş bilgilerin ne kadarının unutulması gerektiğine karar verir. Genel itibarıyla KÖB daha az

işlem gerektiren bir yapıya sahip olup UKSBA'ya göre eğitim süresi daha kısa olabilmektedir. Kapı Özyinelemeli Birim hücre mimarisi ve bağlantıları Şekil 3.28'de görülmektedir.



Şekil 3.28. Kapı Özyinelemeli Birim hücre mimarisi ve bağlantıları [219]

3.4.3. Derin Öğrenmede Kullanılan Hiperparametrelerin Belirlenmesi

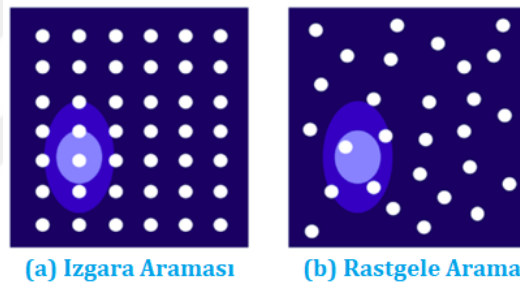
Bir derin öğrenme modeli oluştururken, modelin katman sayısı, her katmandaki nöron veya filtre sayısı, gizli katmanlarda aktivasyon fonksiyonu olarak ReLU'nun veya farklı bir aktivasyon fonksiyonunun kullanılması, belirli bir katmandan sonra BatchNormalization kullanılması, seyreltme (dropout) kullanılması gibi kararların alınması gerekir. Mimari düzeyindeki bu parametrelere, onları bir modelin geri yayılım yoluyla eğitilmiş parametrelerinden ayırt etmek için hiperparametreler denir. Uygulamada, deneyimli makine öğrenimi mühendisleri ve araştırmacıları, bahsedilen tasarım seçenekleri söz konusu olduğunda neyin işe yarayıp neyin yaramadığı konusunda zamanla sezgiler (hiperparametre ayarlama becerileri) geliştirirler [220]. Model parametreleri ile hiperparametreleri farklı kavramlardır. Model tarafından veriden tahmin edilen parametrelere model parametreleri denir. Bir DSA'nın ağırlıkları örnek verilebilir. Model hiperparametreleri ise model tarafından veriden yola çıkarak tahmin edilemeyen parametrelerdir. Bu parametreler, model parametrelerini tahmin etmek için kullanılır. DSA'lardaki öğrenme oranı model hiperparametrelerinden biridir.

Derin öğrenmede belirli sayıda hiperparametre ile problemin eğitim verilerini maksimum düzeyde doğru tahmin edebilecek bir modelleme yapılıır. Model eğitimi sürecinde tasarlanan derin öğrenme mimarisine göre nöronların ağırlıkları öğrenilir. Hiperparametrelerin seçimi başarılı bir model üretme sürecinde kritik rol oynar. Ancak, günümüzde parametrelerin doğru seçilmesine rehberlik edebilecek genel olarak kabul görmüş formüller bulunmamaktadır. Genel olarak ızgara arama ve rastgele arama olmak üzere iki yaklaşım vardır [221].

Izgara arama, arama alanını ayrı bir değerler ızgarasına böler ve ızgaradaki tüm olası kombinasyonları test eder. Örneğin, 3 hiperparametre ve her biri için yalnızca 2 aday değeri olan bir ızgara varsa, toplam 6 kombinasyon kontrol edilmelidir. Izgara araması uygulamak için sınırların belirlenmesi gerekebilir. Izgara araması yüksek boyutlu alanlarda sorun çıkartabilir. Ancak algoritmanın çalıştığı hiperparametre değerleri genellikle birbirinden bağımsız olduğundan,

genellikle kolayca paralelleştirilebilir. Rastgele arama, parametre arama uzayının rastgele bir örneklemesini gerçekleştirir ve bu basit yaklaşımın birçok durumda ızgara aramaya göre daha iyi çalıştığı kanıtlanmıştır [222]. Izgara arama yöntemindeki gibi muhtemel bütün parametre kombinasyonlarının denenmesi yerine rasgele seçilen alt kümeler üzerinden hata değerleri hesaplanır. Hem ızgara araması hem de rastgele arama brute-force yöntemlerdir. Sonraki hiperparametre setini seçmek için önceki sonuçlara bağlı değildirler. Hiperparametre seçimi denemeleri paralel olarak işlenerek avantajlı bir yöntem geliştirilebilir [223].

Şekil 3.29'da verilen örneği inceleyerek rastgele aramanın ızgara aramasından nasıl daha iyi olduğu anlaşılabilir. Şekil 3.29'daki her iki görüntüde de 'x' ve 'y' eksenini iki hiperparametreyi temsil eder ve arka plan, renk açıldıkça artan doğruluğu temsil eder. Şekil 3.29 (a)'da gösterilen ızgara araması durumunda, ızgara boyunca sol üst köşeden aramaya başlanırsa, aramanın daha yüksek doğruluklu bölgeye ulaşması önemli ölçüde zaman alacağı aşikârdır. Şekil 3.29 (b)'de gösterilen rasgele arama durumunda, hiperparametreler rasgele arandığında, ızgara aramasından daha erken ve daha yüksek doğruluğa ulaşma şansı daha yüksektir. Belirlenen deneme sayısı biter bitmez, en iyi kümeye yakın mevcut en iyi hiperparametre kümesini seçilir [223].



Şekil 3.29. (a) Izgara aramasının (b) rastgele aramayla karşılaştırılması [223]

Sinir ağlarında modelin hiperparametreleri olarak adlandırılan öğrenme oranı, yığın, mini-yığın, adım sayısı, katman sayısı, aktivasyon fonksiyonu gibi parametreler doğru olarak seçildiği takdirde istenen başarının elde edilmesi mümkün olmaktadır.

Eğitim verisinin tek seferde hesaplanmasına *yığın-batch*, rastgele seçilmiş eşit büyüklükte, birbirinden bağımsız veri miktarına da *mini-yığın/mini-batch* denir.

Eğitim verisinin tamamının ve geri yayılım ile model ağırlıklılarının güncellenmesinin tamamlanmasına *epoch (adım sayısı)* denir. Hem ileri hem de geri yayılımda tüm eğitim verisinin sinir ağından bir kez geçirilmesi adım sayısı olarak ifade edilmektedir. Ağın eğitimine başlamadan önce kullanılacak olan adım sayısı seçilebilir. Adım sayısı yüksek olursa, ağ daha yüksek doğruluk değerine ulaşabilmektedir. Ayrıca, dikkatli seçilmeyen ve çok yüksek seçilen adım sayısı ağın aşırı uyumuna (overfitting) sebep olabilmektedir. Mini-yığın üzerinden gerçekleştirilen adımlara *iterasyon* denir. Örneğin 256 adet örneğe sahip bir veri setinin kullanıldığı eğitimde mini-yığın boyutunun 8 olarak seçilmesi durumunda 1 adım sayısının tamamlanması için 32 iterasyon gerçekleştirilir.

Gizli katman sayısı, DSA'ların performansını belirleyen bir diğer hiperparametredir. Derin öğrenme mimarilerinde katman sayısı arttırmak her zaman öğrenmenin daha iyi sonuçlanacağı anlamına gelmemektedir. Bunun sebebi katman sayısının gereksiz arttırılması verilerin ezberlenmesine ve test verileri üzerindeki başarımın azalmasına neden olmaktadır. Derin öğrenme modellerinin hiyerarşik olarak otomatik özellik çıkartımı gerçekleştirir. Buradan yola çıkarak problem ve veri türüne bağlı olarak çeşitli özellik seviyelerinin öğrenilmesi sezgisel olarak hedef alındığından öğrenilmesi istenen özellikler dikkate alınarak gerekli katman sayısı tahminle belirlenebilir. Gizli katman sayısı ile katmanlarda bulunan nöron sayısı eşit alınma gibi bir gereklilik olmamakla birlikte farklı seçilebilir.

Öğrenme oranı, maliyet fonksiyonunu minimize etmek için yapılan, her bir iterasyonda elde edilen en aza indirgeme miktarına öğrenme oranı olarak ifade edilmektedir. Problem çözümünde en uygun sonucu elde etmek için öğrenme oranı dikkatli bir şekilde seçilmelidir. Aksi takdirde en uygun çözüm elde edilemeyebilir ve ağırların en uygun çözüme yaklaşması sonsuza dek sürebilir.

Aktivasyon fonksiyonu, seçimi modelin öğrenme yeteneğinde kritik bir rol oynar ve katmanlar arasında farklılık gösterebilir [213].

Seyreltme (Dropout), derin sinir ağlarında sık karşılaşılan problemlerden biri olan aşırı öğrenme (overfitting) probleminin önüne geçmek amacıyla model eğitimi sırasında rastgele seçilen bazı nöronların atılması yöntemidir. Seyreltme, bir çeşit regülerizasyon işlemi gerçekleştirerek tüm nöronların aynı anda öğrenmesi yerine farklı kombinasyonlarla kullanılmasıyla, veri seti üzerinde daha iyi bir genellenimin yapılabilmesini sağlayan etkili bir tekniktir. Geoffrey Hinton ve arkadaşları tarafından aşırı öğrenme probleminin giderilmesi amacıyla önerilmiştir [224].

3.4.4. Derin Öğrenmede Optimizasyon Yöntemleri

Bilim ve mühendislik alanlarındaki birçok problem, parametrelerine göre maksimizasyon veya minimizasyon gerektiren bazı skaler parametrelili amaç fonksiyonlarının optimizasyonu olarak ifade edilebilir. Herhangi bir matematiksel ifadeyi minimize etme veya maksimize etme işlemine optimizasyon denir. Optimize ediciler (optimizers), kayıpları azaltmak için ağırlıklar ve öğrenme oranı gibi sinir ağının özelliklerini değiştirmek ve en uygun değere getirebilmek için kullanılan algoritmalar veya yöntemlerdir. Gradyan iniş, türevlenebilir bir fonksiyonun yerel minimumunu bulmak için birinci dereceden yinelemeli popüler bir optimizasyon algoritmasıdır ve sinir ağlarını optimize etmenin açık ara en yaygın yoludur [225]. Maliyet fonksiyonu kullanılarak elde edilen hata değerinin en aza indirgenmesi için gradyan iniş algoritması kullanılmaktadır. Gradyan iniş yönteminde rastgele alınan değerlerle başlanarak tepeden inişteki küçük adımlarla oluşan en az eğimli hareketle global minimum değerine ulaşım sağlanmaktadır. Derin öğrenme yöntemlerinin popülerleşmesi ve gelişmesiyle birlikte gradyan iniş tabanlı optimizasyon yöntemlerinde yeni

gelişmeler ortaya çıkmıştır. Standart bir gradyan iniş yaklaşımında öğrenme oranı değeri sabit iken yeni algoritmalar sayesinde adaptif olarak kendisini güncellediği çeşitli problemlerde performans iyileşmeleri görülmüştür [225,226].

Amaç fonksiyonunun gradyanını hesaplamak için ne kadar veri kullandığına bağlı olarak farklılık gösteren yığın gradyan inişi, stokastik gradyan inişi ve mini-yığın gradyan inişi olmak üzere üç çeşit gradyan iniş yaklaşımı vardır. Veri miktarına bağlı olarak, parametre güncellemesinin doğruluğu ile bir güncellemeyi gerçekleştirmek için gereken süre arasında bir ödünleşim yapılır. *Yığın gradyan inişi*, tüm eğitim veri kümesi için maliyet fonksiyonunun gradyanını hesaplar. Buna karşılık, *stokastik gradyan inişi* her eğitim örneği ve etiketi için bir parametre güncellemesi gerçekleştirir. *Mini-yığın gradyan inişi* her iki yaklaşımın da en iyisini alır ve tüm eğitim örneklerinin her mini grubu için bir güncelleme gerçekleştirir. Bununla birlikte, bu üç yaklaşım, iyi bir yakınsama garanti etmemekle birlikte ele alınması gereken birkaç problem taşımaktadır. Uygun bir öğrenme oranının seçilememesi, öğrenme oranının bir veri kümesinin özelliklerine uyum sağlayamaması, tüm parametre güncellemeleri için aynı öğrenme oranının geçerli olması, çok sayıda yerel minimumların olması durumunda bu yerel minimumlardan kaçınamama gibi problemlerle karşılaşılabilir. Belirtilen zorluklarla başa çıkmak için derin öğrenme topluluğu tarafından yaygın olarak kullanılan yeni optimizasyon algoritmaları geliştirilmiştir. Bu algoritmalar momentum, Nesterov hızlandırılmış gradyan (Nesterov accelerated gradient), Adagrad, Adadelata, RMSprop, Adam, Adamax ve Nadam gibi yeni yaklaşımlardır [225,227,228]. Stokastik bir gradyan iniş tekniği olan Adam yaklaşımı bu tez çalışmasındaki çoğu derin öğrenme modellerinde tercih edilen optimizasyon algoritmasıdır.

3.4.5. Derin Öğrenme Modellerinin Performans Ölçütleri

Derin öğrenme algoritmalarının sonuçlarının değerlendirilmesi, performansının önemli bir parçasıdır. Model, doğruluk gibi bir metrik kullanılarak değerlendirildiğinde tatmin edici sonuçlar verebilir, ancak diğer metriklerle karşı değerlendirildiğinde istenen sonuçları vermeyebilir. Tasarlanan derin öğrenme modelinin performansını ölçmek için çoğu zaman sınıflandırma doğruluğu kullanılır, ancak bu teknik modelin gerçek anlamda değerlendirilmesi için yeterli değildir.

Bir sınıflandırıcının, gerçek değerleri bilinen bir dizi test verisi üzerindeki performansının değerlendirilmesi için karmaşıklık matrisi kullanılabilir [47,229,230]. Tüm diagonal elemanlar doğru sınıflandırılmış sonuçları gösterir. Yanlış sınıflandırılan sonuçlar, karmaşıklık matrisinin köşegenleri dışında temsil edilir. Bu nedenle, en iyi sınıflandırıcı, yalnızca köşegen elemanları ve geri kalan elemanları sıfıra ayarlanmış bir karmaşıklık matrisine sahip olacaktır. Bir karmaşıklık matrisi, sınıflandırma sürecinden sonra gerçek değerleri ve tahmin edilen değerleri üretir. Sistemin

etkinliđi, matriste oluşturulan Tablo 3.1’deki değere göre belirlenir. Bu tablodan görüleceđi gibi iki sınıf için sınıflandırıcılar karmaşıklık matrisine sahiptir.

Tablo 3.1. Karmaşıklık matrisi tablosu

	Tahmin edilen değeri Negatif	Tahmin edilen değeri Pozitif
Gerçek değeri Negatif	Doğru negatif TN	Yanlış pozitif FP
Gerçek değeri Pozitif	Yanlış negatif FN	Doğru pozitif TP

Karmaşıklık matrisindeki girişler aşağıdaki gibi tanımlanır:

- Doğru pozitif (TP), gerçek sınıf pozitif olduğunda toplam doğru sonuç veya tahmin sayısıdır.
- Yanlış pozitif (FP), gerçek sınıf pozitif olduğunda toplam yanlış sonuç veya tahmin sayısıdır.
- Gerçek negatif (TN), gerçek sınıf negatif olduğunda doğru sonuçların veya tahminlerin toplam sayısıdır.
- Yanlış negatif (FN), gerçek sınıf negatif olduğunda toplam yanlış sonuç veya tahmin sayısıdır.

Doğruluk (Accuracy), sistemin etkinliğini karşılaştırmak için kullanılır. Sınıflandırıcı tarafından yapılan toplam doğru tahmin sayısını hesaba katar. Genel başarı ölçütü olarak kullanılır. Böylece doğruluk

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.16)$$

olur. *Hassasiyet (Recall)*, doğru tanımlanmış pozitif girdilerin oranı alınarak hesaplanır. Doğru pozitif oranıdır ve

$$\text{Hassasiyet} = \frac{TP}{TP+FN} \quad (3.17)$$

ile ölçülür. *Keskinlik (Precision)*, sınıflandırıcı tarafından doğru tahmin edilen pozitif durumlardır. (3.18) ifadesi ile ölçülür.

$$\text{Keskinlik} = \frac{TP}{TP+FP} \quad (3.18)$$

(3.19)'daki *F1-Skoru* (*F1 Score – F measure*), testin doğruluğunun bir ölçüsüdür. Ağırlıklı bir kesinlik ve hassasiyet ortalaması olarak tanımlanır. Aynı zamanda, kesinlik ve hassasiyet arasındaki dengeyi belirtir:

$$F1\ skoru = \frac{2 * Kesinlik * Hassasiyet}{Kesinlik + Hassasiyet} \quad (3.19)$$

3.4.6. Derin Öğrenme Yazılım Araçları

Derin öğrenme algoritmalarını bilgisayar diline çevirmek için birçok yazılım aracı kullanılmaktadır ve algoritmaların uygulanmasında kolaylık sağlamaktadır. Bu yazılım araçları sayesinde derin sinir ağlarının tasarlanmaları, eğitilmeleri ve doğrulanmasının yapabilmelerine imkân tanınmaktadır. Bu yazılım araçları başlıca Torch, Theano, Caffe, TensorFlow, Deeplearning4j, NVIDIA cuDNN, MatConvNet, Keras and PyTorch olarak sıralanabilmektedir [231–237].

Torch: GİB kullanma önceliği olan makine öğrenme algoritmaları için bilimsel hesaplama yapısıdır. Torch; hızlı ve kolay bir kodlama dili olan LuaJIT dili sayesinde ve C/CUDA uygulama temelli, etkili ve kullanımı kolay bir dil yapısına sahip olup açık kaynak kodlu bir kütüphanedir.

Torch; makine öğrenmesi, bilgisayar görmesi, sinyal işleme, paralel işleme, görüntü, video, ses işleme gibi alanlarda geniş bir kullanıcı topluluğu tarafından desteklenen paket programlar sunmaktadır. Karmaşık sinir ağları topolojilerinin uygulanmasında maksimum esnekliğe sahip kullanımı kolay optimizasyon kütüphaneleri ve popüler sinir ağlarını barındırmaktadır. Facebook, Google, Twitter, New York Üniversitesi, Idiap Araştırma Kurumu, Purdue Üniversitesi ve daha birçok şirket ve araştırma laboratuvarları tarafından kullanılmaktadır.

Theano: BYD lisansı altında yayınlanan açık kaynak kodlu bir yapıdır ve MILA grubu tarafından Yoshua Bengio bulunduğu Montreal Üniversitesinde geliştirilmiştir. Adını yunan bir matematikçiden almaktadır. Theano çok boyutlu dizileri içeren matematiksel ifadelerin tanımlanması, optimize edilmesi ve değerlendirilmesine olanak sağlayan ve NumPy kütüphanesini kullanan bir Python kütüphanesidir. 2007'den beri geniş ölçekli hesaplama açısından yoğun bilimsel araştırmaları desteklemektedir. Derin öğrenmedeki sinir ağları algoritmaları için gereken hesaplamaların yapılabilmesi için tasarlanmıştır. Kendi türünün ilk kütüphanelerinden biridir. Derin öğrenme araştırma ve geliştirme alanında endüstriyel standartlardan biridir.

Caffe: Berkeley Görü ve Öğrenme Merkezi ve topluluk katılımcıları tarafından geliştirilmiştir. Yangqing Jia bu projeyi California Berkeley Üniversitesi'nde doktorasını yaparken yapmıştır. BYD lisansı altında yayınlanmıştır. İfade edilebilir bir mimari yapısına sahip olması sebebi ile farklı uygulamalar ve inovasyonları desteklemektedir. Optimizasyon ve modeller kodda herhangi bir değişiklik yapma gereği duyulmadan konfigürasyon ile tanımlanabilmektedir. Genişletilebilir bir kod olması aktif gelişimine katkıda bulunmuştur. Caffe'nin ilk yılında 1000'in

üzerinde geliştirici tarafından kullanılmıştır ve bu geliştiriciler Caffé'nin gelişimine çok önemli katkılar sağlamışlardır. Hızlı olması endüstriyel uygulamalar ve araştırma deneylerinde Caffé'yi tercih sebebi yapmaktadır. Akademik araştırma projeleri, başlangıç prototiplerini ve hatta görme, konuşma ve multimedya alanlarında geniş ölçekli endüstriyel uygulamaları desteklemektedir.

TensorFlow: Veri akış grafikleri kullanarak numerik hesaplamalar yapan açık kaynak yazılım kütüphanesidir. Grafikteki düğümler matematiksel işlemleri temsil etmektedir. Grafik kenarları ise grafikler arasındaki bağlantıyı sağlayan çok boyutlu veri dizilerine karşılık gelmektedir. TensorFlow, esnek bir yapıya sahiptir. TensorFlow başlangıçta Google Machine Intelligence grubu içerisinde faaliyet gösteren Google Beyin takımında çalışan araştırmacılar ve mühendisler tarafından makine öğrenmesi ve DSA araştırmalarını birleştirmek amacıyla geliştirildi. Şuan da diğer birçok alanda da genel olarak uygulanabilir seviyeye kavuşmuş bulunmaktadır.

Deeplearning4j (DL4J): Java ve Scala için yazılmış açık kaynak dağıtık derin öğrenme kütüphanesidir. Hadoop ve Spark ile entegre olan DL4J dağıtık MİB ve GİB'ler üzerinde iş ortamlarında kullanılmak üzere tasarlanmıştır. Ticari desteği Skymind şirketi tarafından sağlanmaktadır. Apache 2.0 lisansı altında yayınlanmıştır. DL4J dolandırıcılık tespiti, ağ ihlali tespiti, yüz/imge tanıma, ses arama, sesi metine dönüştürme gibi alanlarda kullanılabilir. DL4J sinir ağı modellerini Keras, TensorFlow, Caffé, Torch ve Theano gibi en çok gelişmiş sistemlerden aktarabilmektedir. DL4J'lerin esnekliği Spark ve Hadoop ile MİB ve GİB'ler üzerinde çalışan dağıtık ve üretim nitelikli bir yapıda ihtiyaç duyulduğu gibi KBM, diğer otomatik kodlayıcılar, KSA ya da özyinelemeli ağları birleştirebilme olanağı sunmaktadır.

NVIDIA cuDNN (The NVIDIA CUDA Deep Neural Network library): DSA için GİB hızlı kütüphanedir. cuDNN; ileri ve geri konvolüsyon, örnekleme, normalizasyon ve aktivasyon katmanları gibi standart rutinler için yüksek ayarlı uygulamaları sağlamaktadır. cuDNN NVIDIA derin öğrenme yazılım geliştirme kitinin bir parçasıdır. Derin öğrenme araştırmacıları sistem geliştiricileri yüksek performanslı GİB hızlandırma için cuDNN'e ihtiyaç duymaktadır. cuDNN'de düşük seviyeli GİB performansı ayarlamada zaman harcamaktan ziyade sinir ağlarını eğitmeye ve yazılım uygulamaları geliştirmeye odaklanılmıştır. cuDNN; caffe, torch, theano ve tensorflow gibi yaygın olarak kullanılan derin öğrenme yapılarını hızlandırabilmektedir. Veri bilimcileri ve araştırmacıları derin öğrenme yapılarını ya da NVIDIA DIGITS indirerek cuDNN'den faydalanabilmektedir. DIGITS interaktif olarak veri yönetebilme ve eğitimi çoklu GİB'ler üzerinde gerçekleştirebilme olanağı sunmaktadır.

MatConvNet: Bilgisayarlı görü için KSA'ları kullanan bir MATLAB Toolbox'dır. MatConvNet bilgisayarlı görü alanında ve imge anlamada bilhassa VLFeat açık kaynak kütüphanesi tarafından kullanılan geleneksel imge gösterimlerinin yerini alma gibi önemli gelişmeler göstermiştir. KSA'nın çoğu konvolüsyon ve düzeltme gibi basit lineer ve lineer olmayan

filtreleme işlemlerini oluşturarak elde edilmesine karşılık uygulamaları önemlidir. Bunun sebebi KSA'nın çok büyük miktardaki veriden öğrenme ihtiyacıdır. KSA kütüphanelerinin çoğunda olduğu gibi MatConvNet bunu farklı optimizasyon tekniklerini kullanarak ve genellikle GİB'ler üzerindeki hesaplamaları destekleyerek elde etmektedir. MatConvNet BYD lisansı altında yayınlanan açık kaynaklardan biridir. MatConvNet; konvolüsyon, normalizasyon ve örnekleme gibi KSA oluşturma bloklarını MATLAB komutlarına benzer şekilde basit bir dil yapısı ile sunmaktadır. KSA mimarilerinin oluşturulması için bu bloklar bir araya getirilip genişletilebilmektedir. Benzer blokların çoğu C++ ve CUDA ile yazılan optimize edilmiş MİB ve GİB uygulamalarını kullanmaktadır. Diğer taraftan GİB hesaplama yönelik MATLAB'ın doğal desteği, MATLAB ortamında yeni blokların yazılmasının hesaplamayla ilgili verimliliği korumak ile mümkün olduğu anlamına gelmektedir. Düşük seviyeli diller kullanarak yeni KSA bileşenlerini yazmaya karşılık, yeni fikirleri test etmeyi önemli ölçüde hızlandırıp basitleştirmektedir. MATLAB kullanımı diğer alanlar ile ortak çalışma ortamı hazırlamaktadır.

PyTorch: PyTorch, DSA'ları modellemek ve hesaplamaları çalıştırmak için kullanılan Torch derin öğrenme çerçevesinden geliştirilmiştir. PyTorch açık kaynak kitaplıktır. PyTorch, Python üzerinde çalışır ve GİB üzerinde iyi eğitilmiştir. PyTorch, hesaplama grafiğini kullanır ve kullanıcıların kendi başlarına çerçeve oluşturmaya ve çalışma zamanı sırasında bunları değiştirmesine olanak tanır. PyTorch, esnek ve dile daha yakın olduğu için zorunlu stil derin öğrenme çerçevesidir. Bu, bellek tahsisi kullanıcı gereksinimlerine göre yapıldığından değerlidir. CUDA uygulamasını destekler ve hızlı, taşınabilir ve kullanımı kolaydır. Derin öğrenme algoritmaları için birçok kütüphaneye sahiptir. Sinir ağları, görüntü işleme ve enerji tabanlı modeller için paketler içerir. Konuşma, video ve görüntü uygulamalarını içeren uygulamaları vardır. ÖSA, KSA vb. ağlar PyTorch kullanılarak eğitilebilir. PyTorch, iOS ve Android'e yerleştirilebilir. Google, Twitter, Nvidia vb. tarafından kullanılır.

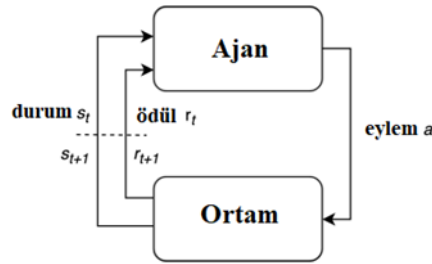
Keras: Python ile yazılmış açık kaynak kodlu bir kütüphanedir. Öncelikle Sinir Ağları oluşturmak için kullanılır ve arayüzü boyunca sürekli olarak daha temiz bir çerçeve ile sonuçlanan nesne yönelimli bir tasarım izler. Keras, TensorFlow, Theano, MXNet ve CNTK gibi çoklu arka uç kaynaklarıyla çalışma yeteneğine sahiptir. 2015'in başlarında tanıtılmıştır. Diğer çerçevelerle bağlantılı olarak çalışır ve çeşitli üst düzey API'ler ve görüntü ve metin işlemeyi basitleştiren bir dizi araç sağlar.

3.5. Derin Pekiştirmeli Öğrenmeye Genel Bir Bakış

Pekiştirmeli öğrenme tekniği, makine öğrenmesinin denetimli öğrenme ve denetimsiz öğrenme yaklaşımlarından farklıdır. Denetimli öğrenmede; etiketli bir veri setinden eğitim ve test kümeleri oluşturulur. Denetimli öğrenme eğitim kümesi üzerinde oluşturulan model ile bu modelin performansının (öğrenme düzeyinin) test kümesi üzerinden incelenmesine dayanmaktadır. Bu

sayede yeni gelen etiketsiz örnekler, model yardımıyla etiketleri tahmin edilebilir hale getirilmektedir. Denetimsiz öğrenme ise genellikle etiketlenmemiş veriler üzerinde kümeleme tekniği kullanılarak yapılmaktadır. Pekiştirmeli öğrenme, sıralı karar verme problemlerini çözmekle ilgilenir. Video oyunları oynama, spor yapma, araç kullanma, envanteri optimize etme, robotik kontrol gibi birçok gerçek dünyada karşılaşılan sorun bu çerçevede ele alınabilir.

Pekiştirmeli öğrenme, optimal kontrol teorisine ve MKS'ye kadar uzanan bir yapay zeka alt alanıdır. İlk olarak 1950'lerde dinamik programlama ve yarı doğrusal denklemler bağlamında Richard Bellman tarafından ele alınmıştır [238,239]. MKS algılama (sensation), eylem (action) ve hedef (goal) olmak üzere üç en önemli özelliğe sahiptir [240–242]. Pekiştirmeli öğrenme psikolojideki davranışçı yaklaşımdan esinlenen, ajanın-karar vericinin (agent) bir ortamda (environment) en yüksek ödül (cumulative reward) değerine ulaşabilmesi için hangi eylemleri yapması gerektiğine karar verebilen bir mekanizmanın oluşturulmasıyla ilgilenen bir makine öğrenmesi tekniğidir [243]. Pekiştirmeli öğrenme problemleri, bir ajan ve bir ortamdan oluşan bir sistem olarak ifade edilebilir. Ortam, sistemin durumunu tanımlayan bilgileri üretir ve bu da durum olarak tanımlanır. Ajan, ortamı gözlemleyerek ve bu bilgiyi bir eylem seçmek için kullanarak ortamla etkileşime girer. Ortam eylemi kabul eder ve bir sonraki duruma geçer. Daha sonra bir sonraki durumu ve ödülü ajana verir. (Durum --> Eylem --> Ödül) döngüsü tamamlandığında, bir zaman adımının(time step) geçtiği kabul edilir. Döngü, örneğin problem çözüldüğünde, ortam sona erene kadar tekrar eder. Tüm bu süreç, Şekil 3.30'daki pekiştirmeli öğrenme diyagramı ile açıklanmıştır.



Şekil 3.30. Pekiştirmeli öğrenme süreci [243]

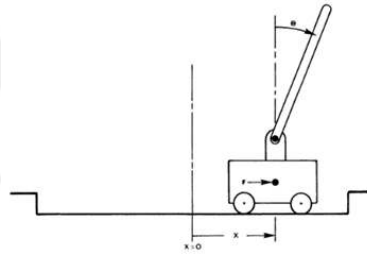
Bir ajanın eylem üretme işlevine politika denir. Politika, durumları eylemlerle eşleştiren bir işlevdir. Ajanın yaptığı her eylemin ortamı değiştirmesinden dolayı ajanın daha sonra ne gözlemlediğini ve ne yaptığını da etkilemektedir. Ajan ve ortam arasındaki alışveriş zamanla ortaya çıkar ve bu nedenle Pekiştirmeli öğrenme sıralı bir karar verme süreci olarak düşünülebilir.

Pekiştirmeli öğrenme problemlerinde amaç, ajanın iyi eylemleri seçerek almış olduğu ödüllerin toplamını en üst düzeye çıkarmaktır. Bunu da, deneme yanılma sürecinde ortam ile etkileşime girerek öğrenir ve aldığı ödül sinyallerini iyi eylemleri pekiştirmek için kullanır. Temel olarak, pekiştirmeli öğrenme sistemi, ajanın ortam ile etkileşime girdiği ve sinyal alışverişinde

bulunduğu, aynı zamanda amacını (ödülü) maksimize etmeye çalıştığı bir geri beslemeli kontrol döngüsüdür.

Değiştirilen sinyaller (s_t ; a_t ; r_t), sırasıyla durum, eylem ve ödül anlamına gelir ve t , bu sinyallerin meydana geldiği zaman adımını belirtir. (s_t ; a_t ; r_t) demetine deneyim denir. Kontrol döngüsü bir terminal durumuna veya maksimum zaman adımını $t = T$ 'ye ulaşarak sonlandırılabilir. $t=0$ 'dan ortamın sona erdiği zamana kadar olan sürece bölüm(episode) denir. Yörünge (trajectory), bir bölüm süresince gerçekleştirilen deneyimlerin bir dizisidir. $\tau = (s_0, a_0, r_0), (s_1, a_1, r_1), \dots, (s_T, a_T, r_T)$ şeklinde ifade edilir. Ajan genellikle, iyi bir politika öğrenmek için, sorunun karmaşıklığına bağlı olarak, yüzlerce ile milyonlarca arasında değişen birçok bölüme ihtiyaç duyar. Şekilde gösterilen örnek bir pekiştirmeli öğrenme ortamında durumların, eylemlerin ve ödüllerin nasıl tanımlandığına bakalım. Cartpole ortamı, standartlaştırılmış ortamlar sağlayan açık kaynaklı bir kütüphane olan OpenAI Gym aracılığıyla kullanılabilir.

CartPole, ilk olarak 1983'te Barto, Sutton ve Anderson [244] tarafından tanımlanan en basit pekiştirmeli öğrenme ortamlarından biridir. Şekil 3.31'de Cartpole sisteminin görsel hali görülmektedir.



Şekil 3.31. Cartpole sistemi [244]

Ortamda, sürtünmesiz bir yol boyunca hareket ettirilebilen bir arabaya tutturulan direğin dengede tutulması istenir. Ortamın ana özellikleri aşağıda özetlenmiştir:

Amaç: Direğin belirli bir süre boyunca dik tutulması.

Durum: [araba konumu, araba hızı, kutup açısı, kutup açısal hızı] temsil eden bir dizi. Örneğin, $[-0.034, 0.032, -0.031, 0.036]$.

Eylem: Bir tamsayı, ya arabayı sola sabit bir mesafeye taşımak için 0 ya da arabayı sağa sabit bir mesafeye taşımak için 1.

Ödül: Direk dik kaldığı her adım için +1.

Sonlandırma: Direk devrildiğinde (dikeyden 12 dereceden fazla) veya araba ekrandan çıktığında veya maksimum zaman adımına ulaşıldığı an.

3.5.1. Markov Karar Süreçleri

MKS, tüm pekiştirmeli öğrenme süreçlerinin temelini teşkil etmektedir [111]. Bir ortam bir durumdan diğerine geçtiğinde bir *geçiş işlevi* kullanılır. Geçiş fonksiyonlarının neden MKS

olarak temsil edildiğini anlamak için 3.20'de verilen genel formülasyon göz önünde bulundurulabilir.

$$s_{t+1} \sim P(s_{t+1} | (s_0, a_0), (s_1, a_1), \dots, (s_t, a_t)) \quad (3.20)$$

Burada; t zaman adımında, bir sonraki durum s_{t+1} 'in tüm geçmişe bağlı bir olasılık dağılımı olan P'den örneklendiğini ifade eder. Bir ortamın s_t durumundan s_{t+1} 'e geçme olasılığı, bir bölümde (episode) o ana kadar meydana gelen önceki tüm durumlara ve eylemlere bağlıdır. Bu formda bir geçiş fonksiyonunu modellemek, özellikle de bölümler birçok zaman adımından oluşuyorsa zordur. Tasarladığımız herhangi bir geçiş fonksiyonunun, geçmişte herhangi bir noktada meydana gelen geniş bir etki kombinasyonunu hesaba katabilmesi gerekir. Ayrıca, bu formülasyon, bir ajanın eylem üretme işlevini, yani politikasını, önemli ölçüde daha karmaşık hale getirir. Durumların ve eylemlerin tüm geçmişi, bir eylemin ortamın gelecekteki durumunu nasıl değiştirebileceğini anlamakla ilgili olduğundan, bir ajanın nasıl hareket edeceğine karar verirken tüm bu bilgileri dikkate alması gerekir. Ortam geçiş fonksiyonunu daha pratik hale getirmek için, sonraki s_{t+1} durumuna geçişin yalnızca önceki durum s_t ve önceki eylem a_t 'ye bağlı olduğu varsayımı eklenerek bir MKS'ye dönüştürülebilir. Buna Markov özelliği denir. Bu varsayımla, yeni geçiş fonksiyonu

$$s_{t+1} \sim P(s_{t+1} | s_t, a_t) \quad (3.21)$$

olur. Burada, bir sonraki durum olan s_{t+1} 'in $P(s_{t+1} | s_t, a_t)$ olasılık dağılımına göre belirlendiğini ifade eder ve orijinal geçiş fonksiyonunun daha basit bir şeklidir. Markov özelliği, t adımındaki mevcut durum ve eylemin, t+1'deki bir sonraki durum için geçiş olasılığını tam olarak belirlemek için yeterli bilgiyi içerdiği anlamına gelir. Oyunlar, robotik kontrol ve planlama dahil olmak üzere birçok süreç bu formda ifade edilebilir. Bunun nedeni, Markov geçiş fonksiyonunu elde etmek için gerekli herhangi bir bilgiyi içerecek şekilde bir durumun tanımlanabilmesidir. MKS, S , A , $P(\cdot)$, $R(\cdot)$ şeklinde dördümlü bir demet (tuple) ile tanımlanır.

- S durum kümesidir.
- A , eylemler kümesidir.
- $P(s_{t+1} | s_t, a_t)$, ortamın durum geçiş fonksiyonudur.
- $R(s_t, a_t, s_{t+1})$ çevrenin ödül işlevidir.

Bir ajan geçiş ve ödül fonksiyonları hakkında bilgi alabilmesinin tek yolu, çevrede deneyimlediği durumlar, eylemler ve ödüllerdir. MKS probleminin formülasyonunun tamamlanması için, bir ajanın maksimize ettiği amaç (objective) kavramının da açıklanmasında fayda vardır. İlk olarak bir *bölümlük (episode) yörüngeden (trajectory)* $\tau = (s_0, a_0, r_0), \dots, (s_T, a_T, r_T)$ toplam ödülün (return, $R(\tau)$)

$$R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots + \gamma^T r_T = \sum_{t=0}^T \gamma^t r_t \quad (3.22)$$

gibi tanımlanır. Burada, toplam ödül bir yörüngedeki ödüllerin indirimli toplamı olarak tanımlar. Burada γ indirim faktörüdür. $J(\tau)$ amacı, birçok yörüngeden elde edilen toplam ödüllerin beklentisi olup

$$J(\tau) = \mathbb{E}_{\tau \sim \pi}[R(\tau)] = \mathbb{E}_{\tau}[\sum_{t=0}^T \gamma^t r_t] \quad (3.23)$$

gibidir. Toplam ödül $R(\tau)$, $t = 0, \dots, T$ tüm zaman adımlarında $\gamma^t r_t$ indirimli ödüllerin toplamıdır. $J(\tau)$, birçok bölüm üzerinden ortalama toplam ödüdür. Beklenti, eylemlerdeki ve ortamdaki stokastikliği açıklar. Yani, tekrarlanan çalışmalarda, toplam ödül her zaman aynı olmayabilir. Amacı maksimize etmek, toplam ödülü maksimize etmekle aynıdır.

$\gamma \in [0,1]$ indirim faktörü, gelecekteki ödüllerin değerlendirilme şeklini değiştiren önemli bir değişkendir. γ ne kadar küçükse, gelecekteki zaman adımlarında ödüllere o kadar az ağırlık verilir. $\gamma = 0$ durumunda, amaç yalnızca ilk ödül r_0 'ı dikkate alır ve

$$R(\tau)_{\gamma=0} = \sum_{t=0}^T \gamma^t r_t = r_0 \quad (3.24)$$

gibi olur. γ ne kadar büyükse, gelecekteki zaman adımlarında ödüllere o kadar fazla ağırlık verilir. $\gamma = 1$ ise, her zaman adımından elde edilen ödüller eşit olarak

$$R(\tau)_{\gamma=1} = \sum_{t=0}^T \gamma^t r_t = \sum_{t=0}^T r_t \quad (3.25)$$

ile ağırlıklandırılır. Bir MKS olarak formüle edilmiş pekiştirmeli öğrenme ile ajan, politika olarak bilinen eylem üreten bir fonksiyonu öğrenebilir. Ancak, bir ortamın bir ajan için yararlı olabilecek başka özellikleri de vardır. Özellikle pekiştirmeli öğrenmede öğrenilmesi gereken üç temel işlev bulunmaktadır:

- Durumu eyleme eşleyen bir politika, $\pi: a \sim \pi(s)$
- Beklenen toplam ödül tahmin etmek için bir değer fonksiyonu, $V^\pi(s)$ veya $Q^\pi(s, a)$, $\mathbb{E}_t[R(\tau)]$
- Ortam modeli, $P(s' | s, a)$

Politika, bir ajanın amacını maksimize etmesi için ortamda nasıl eylemler ürettiği ile ilgilidir. Politika stokastik olabilir. Yani, aynı durum için olasılıksal olarak farklı eylemler üretebilir. Belirli bir durumdaki bir eylemin olasılığını belirtmek için $\pi(a | s)$ ifadesi kullanılabilir. Bir politikadan örneklenen bir eylem $a \sim \pi(s)$ olarak yazılır. Değer fonksiyonları, amaç hakkında bilgi sağlar. Ajanın, gelecekteki beklenen toplam ödül açısından durumların ve mevcut eylemlerin ne kadar iyi olduğunu anlamasına yardımcı olurlar. $V^\pi(s)$ ve $Q^\pi(s, a)$ sırayla Denklem 3.26 ve 3.27 şeklinde ifade edilirler.

$$V^\pi(s) = \mathbb{E}_{s_0=s, \tau \sim \pi}[\sum_{t=0}^T \gamma^t r_t] \quad (3.26)$$

Denklem 3.26'da gösterilen V^π değer fonksiyonu, bir durumun ne kadar iyi veya kötü olduğunu değerlendirir. V^π , ajanın mevcut π politikasına göre hareket etmeye devam ettiğini varsayarak, s durumunda olmaktan beklenen toplam ödül hesaplar. Toplam ödül, $R(\tau) = \sum_{t=0}^T \gamma^t r_t$ olarak mevcut s durumundan bir bölümün sonuna kadar hesaplanır. s durumundan önce alınan tüm ödüller göz ardı edildiğinden ileriye dönük bir önlemdir.

$$Q^\pi(s, a) = \mathbb{E}_{s_0=s, a_0=a, \tau \sim \pi} [\sum_{t=0}^T \gamma^t r_t] \quad (3.27)$$

Denklem 3.27'de gösterilen Q -değer fonksiyonu Q^π , bir durum-eylem çiftinin ne kadar iyi veya kötü olduğunu değerlendirir. Q^π , ajanın mevcut π politikasına göre hareket etmeye devam ettiğini varsayarak, s durumundaki a eyleminden beklenen toplam ödül hesaplar. V^π ile aynı şekilde, toplam ödül, mevcut s durumundan bir bölümün sonuna kadar ölçülür.

3.5.2. Bellman Denklemi

Bellman denklemi, Richard Bellman tarafından 1954 yılında yayınlanan makale ile önerilmiştir [238,239]. MKS problemlerine yinelemeli bir çözüm sunmaktadır. MKS'nin yinelemeli şekli, dinamik programlama ve doğrusal programlama gibi tekrarlı algoritmalar kullanılarak MKS'yi çözmek için kullanılmaktadır. MKS'nin yinelemeli şekli, pekiştirmeli öğrenme ajanının eğitilebilmesi için kullanılan algoritmaların matematiksel temellerini şekillendiren farklı yapılar için bir yaklaşımdır. Bellman denklemi, değer fonksiyonu ve eylem-değer fonksiyonunun matematiksel çözümünün nasıl yapıldığını göstermektedir. Deterministik ortamların ifade edilmesinde kullanılan Bellman denklemi

$$V(s) = \max_a (R(s, a) + \gamma V(s')) \quad (3.28)$$

şeklindedir. Stokastik ortamı içeren değer fonksiyonunun Bellman denklemi ise

$$V^\pi(s) = \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^\pi(s')] \quad (3.29)$$

gibi ifade edilir. Diğer yandan, politika stokastik bir politikaysa, bir olasılık dağılımına dayalı olarak eylemler seçilir; yani, bir durumda aynı eylemi gerçekleştirmek yerine, eylem uzayı üzerindeki olasılık dağılımına göre bir eylem seçimi yapılır. Böylece, stokastik politikayı içeren değer fonksiyonunun Bellman denklemi

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^\pi(s')] \quad (3.30)$$

gibi olur.

3.5.3. Pekiştirmeli Öğrenme Terminolojisi

Pekiştirmeli öğrenmenin daha iyi anlaşılması için bazı temel unsurlardan bahsetmek gerekir. Bu kavramlar ajan-model, ortam, durum, eylem/aksiyon, ödül, politika, indirim faktörü, değer fonksiyonu ve yörünge olarak sıralanabilir [245].

Ajan: Ortamla sürekli etkileşim halinde olup öğrenme sürecinde ortamdan aldığı ödülleri maksimuma çıkarmayı amaçlayan öğrenme aktörüdür.

Ortam: Ajanın içerisinde hareket ettiği ve buna karşılık ajana yanıt verilen dünyaya denir. Ortam ajanın o anki durumunu ve eylemini giriş olarak alır ve ajana ajanın sonraki durumunu ve ödülünü çıkış olarak geri döndürür. Bir örnek vermek gerekirse, insanın ajan olduğunu düşünelim. Bu durumda çevre, insanın yaşadığı dünyadaki davranışlarını işleyen ve sonuçlarını insana yansıtan fizik kanunları ve toplumun kurallarıdır. Ajanın doğrudan gözlemlediği ortama tam gözlemlenebilir, dolaylı olarak gözlemlediği ortama ise kısmi gözlemlenebilir ortam denir.

Durum: Ajanın kendisini içerisinde bulduğu, belirli bir konum ve anın, somut ve gecikmesiz haline denir. Ayrıca ortamda bulunan tüm diğer unsurları da kapsar. Durum (state) ve gözlem (observation) olarak iki farklı şekilde karşımıza çıkmaktadır. Durum genellikle yapıdaki tüm unsurları içerisinde barındırır. Gözlem ise durumun herhangi bir andaki olası rasgele bir fonksiyonudur.

Eylem: Ajanın gerçekleştirebileceği olası tüm eylemler veya alabileceği kararlar dizisidir. Eylem uzayı, ayrık ve sürekli olmak üzere iki çeşittir. Eylem uzayı, ayrık eylemlerden oluştuğunda ayrık eylem uzayı, sürekli olan eylemlerden oluştuğunda sürekli eylem uzayı olarak tanımlanır.

Ödül: Ajanın bir durumdaki eylemlerinin başarılı ya da hatalı olduğunu ölçen bir geri beslemedir. Ajan, izlemiş olduğu politikayı bu ödülleri esas alarak zaman içerisinde değiştirir. Örneğin yapılan bir eylem sonrasında düşük bir puan elde ediliyorsa, gelecekte ajan aynı duruma geldiğinde farklı bir eylem yapmayı tercih edebilir. Başka bir deyişle, ödül ajanın eylemlerinin etkin bir şekilde değerlendirilmesidir. $r_t = R(s_t, a_t)$ ya da $r_t = R(s_t)$ şeklinde gösterilir. Ödül toplamı getiri (toplam ödül, return) olarak ifade edilmektedir.

Politika: Ajanın anlık durumuna göre sonraki eylemini belirlemek için kullanmış olduğu stratejidir. Durumları en yüksek ödülü aldırabilecek eylemlerle eşleştirmektedir. Bir nevi etki-tepki eşleşmesi olarak düşünülebilir. İçinde bulunulan durum bir etki olarak kabul edilirse ajan buna karşılık bir tepki (action) verir veya eylem gerçekleştirir. Bu politika basit bir aksiyon ya da eylem olarak tanımlanabileceği gibi bütün durumları karşılayan bir arama tablosu şeklinde de tanımlanabilir. Politika dinamik olarak da nitelenebilir. Bunun temel nedeni, ajanın içinde bulunduğu durumu değerlendirerek gerçekleştirebileceği eylemleri aramasından kaynaklanmaktadır. Politika, deterministik ve stokastik olarak sınıflandırılabilir. Deterministik politika, ajana bir durumda belirli bir eylemi gerçekleştirmesini söyler. Böylece, deterministik politika, durumu belirli bir eyleme eşler ve genellikle μ ile gösterilir. Deterministik politika, $a_t =$

$\mu(s_t)$ şeklinde ifade edilir. Stokastik bir politika, bir durumu doğrudan belirli bir eyleme eşlemez; bunun yerine, durumu bir eylem alanı üzerindeki bir olasılık dağılımına eşler. Stokastik bir politika, $a_t \sim \pi(s_t)$ veya $\pi(a_t|s_t)$ şeklinde ifade edilir. Stokastik politika, kategorik ve gauss politikası olmak üzere iki türe ayrılabilir. Eylem uzayı ayrık olduğunda, stokastik bir politikaya kategorik bir politika denir. Eylem uzayı sürekli olduğunda, stokastik bir politikaya Gauss politikası denir.

İndirim faktörü: Yunan alfabesi harflerinden olan küçük γ ile gösterilir. Ödül değeri hesaplanırken kullanılır. İndirim faktörü, ajan tarafından uzun vadede keşfedilen ödüller ile bu ödüllerin ajanın aksiyon seçimindeki etkisini azaltmak için çarpılır. Bunun sebebi uzun vadede elde edilen ödüllerin etkisinin anlık alınan ödüllerin etkisinden daha az olmasıdır. Bu durum, ajanda kısa vadeli bir haz halinin oluşmasına benzetilebilir. Gerçek hayatta hepimiz uzun vadede bir ölüyüz, dolayısıyla yaşam yolculuğumuzun başlangıcında yapmış olduklarımız daha değerlidir diyebiliriz.

Değer: Kısa vadeli ödülün aksine belirli bir politikanın etkisinde uzun vadeli istenen getiriye denir. Getiri G_t ile gösterilir ve

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (3.31)$$

gibi yazılır.

Değer fonksiyonu: Ajanın bulunduğu durumun ne kadar iyi olduğunu gösterir. Durum-değer fonksiyonu ve eylem-değer fonksiyonu olmak üzere ikiye ayrılır. Durum-değer fonksiyonu $V_{\pi}(s_t)$ ile gösterilir. Eylem-değer fonksiyonu ise $Q_{\pi}(s_t, a_t)$ ile gösterilir. Ajanın gecikmesiz aldığı ödül sinyalinin aksine, durum fonksiyonu ajanın uzun vadede neyin iyi olduğuna karar vermesine yardımcı olur.

Durum-değer fonksiyonu denklemini olan $V_{\pi}(s_t)$ basitçe π politikasına göre s_t durumunda beklenen getiriyi

$$V_{\pi}(s) = \mathbb{E}_{\pi}[G_t | S_t = s] \quad (3.32)$$

ile hesaplar. Eylem-değer fonksiyonu ise π politikasına göre s_t durumunda başlayan ve a_t aksiyonunu alınmasını hesaplayan beklenen getiri olup

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[G_t | S_t = s, A_t = a] \quad (3.33)$$

ile hesaplanır.

Yörünge (Trajectory): Bir dizi durum ile bu durumları etkileyen aksiyonlardan oluşur ve $\tau = (s_0, a_0, s_1, a_1, \dots)$ ile ifade edilir.

Aralıklı ve sürekli görevler: Pekiştirmeli öğrenmede görevler, aralıklı (episodik) ve sürekli görev olmak üzere iki şekilde kategorize edilir. Aralıklı görev, terminal/son durumu olan bir görevdir. Örneğin, bir araba yarışı oyununda, başlangıç noktasından (ilk durum) başlanır ve

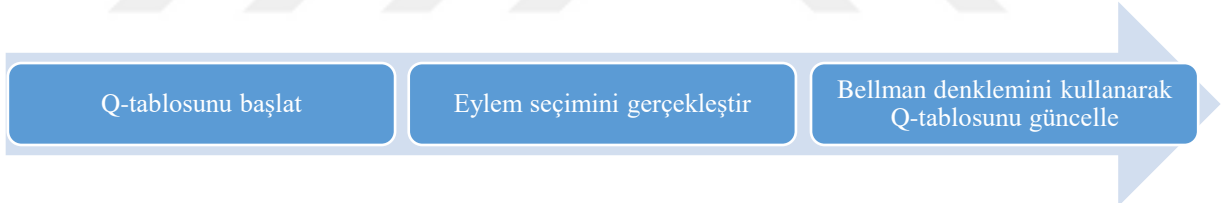
varış noktasına (terminal durumu) ulaşılır. Aralıklı görevlerden farklı olarak, sürekli görevler herhangi bir bölüm (episode) içermez ve bu nedenle herhangi bir terminal/son durumu yoktur. Örneğin, kişisel bir yardım robotunun bir terminal durumu yoktur.

3.5.4. Q-Öğrenme Algoritması

Q-öğrenme, Q fonksiyonu kavramına dayanmaktadır [246,247]. Q-öğrenme de ajana ait bulunduğu durumlar ve gerçekleştirmiş olduğu eylemler çiftine ait veriler içeren bir tablodan faydalanılır. Öğrenme sürecinin her bir adımında Q-tablosunda yer alan Q-değerleri güncellenmektedir. Ajan tabloyu kullanarak bulunduğu durumdaki en uygun eylemi seçmektedir. Q-öğrenme model içermeyen değere dayalı bir pekiştirmeli öğrenme yöntemi olup pozitif ve negatif ödüller ile durum aksiyon çiftini kullanarak öğrenme olayını gerçekleştirmektedir. Pekiştirmeli öğrenmede amaç değer fonksiyonu olan Q-fonksiyonunu

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3.34)$$

ile maksimize etmektir. Bu fonksiyon, Bellman denklemini kullanarak $Q(s_t, a_t)$ 'yı yinelemeli olarak günceller. $Q(s_t, a_t)$, herhangi bir durumdaki eylemin beklenen gelecekteki ödülünü verir. α öğrenme katsayısı ve γ indirim faktörüdür. Temel bir Q-öğrenme sürecinin adımları Şekil 3.32'de görülmektedir.



Şekil 3.32. Temel bir Q-öğrenme sürecinin adımları

Q-öğrenme algoritmasının aşamalarına bakıldığında ilk olarak Q-tablosu başlatılır. Daha sonra epsilon greedy keşif stratejisine dayalı olarak bir eylem seçimi gerçekleştirilir. Son olarak Bellman denklemi kullanılarak Q-tablosunda yer alan değerler güncellenir. Q-tablosu durum-eylem çiftlerini ve onlara karşılık gelen Q-değerlerini bir başka deyişle beklenen ödülleri içermektedir. Q-öğrenme algoritması başlangıçta Q-tablosundaki tüm değerleri sıfır kabul edilerek başlatılır. Bu da ajanın dünya ile ilgili hiçbir şey bilmediğini göstermektedir. Ajan deneme yanılma yoluyla farklı durumlarda farklı eylemleri gerçekleştirerek her bir durum-eylem çiftine ait beklenen ödül öğrenilir ve Q-tablosu yeni Q-değerleriyle güncellenir. Ortamın öğrenilmesinde deneme yanılma yönteminin kullanılmasına keşif (exploration) denir.

Q-öğrenme algoritmasının en önemli amaçlardan biri de yeni ortam için Q-değerlerinin öğrenilmesidir. Q-değeri, bir ajanın verilen bir s durumundan a eylemini gerçekleştirerek ulaşabileceği beklenen maksimum ödüdür. Ajanın her bir durum-eylem çiftine ait Q-değerini

öğrenmesiyle s durumundaki bir ajan en yüksek beklenen ödüllü a eylemini gerçekleştirerek beklenen ödülünü maksimize eder. Bir durumdaki bilinen en iyi ödülün açık bir şekilde seçilmesine sömürü (exploitation) denir.

Pekiştirmeli Öğrenme sürecindeki en önemli zorluklar, keşif (exploration) ve sömürü (exploitation) kavramlarının uygulamaya geçirilmesidir. Ajanın daha fazla ödül elde etmesi için geçmişte denediği ve pozitif ödül aldığı eylemleri seçmelidir. Ajan ödül elde etmek için daha önce deneyimlediği eylemlerden faydalanır. Ancak karşılaştığı bir durumda daha fazla ödül alabileceği eylemler varsa bunları da keşfetmelidir. Böylece ajan, çeşitli eylemler denedikten sonra en iyi ödül alabildiklerini sonraki aşamalarda desteklemelidir. Aksiyon seçimi yapılırken epsilon greedy keşif stratejisi, keşif sömürü ödünleşimine getirilen genel yaklaşımlardan biridir. Algoritmanın başlangıcında ajanın almış olduğu her bir adım rastgele olmaktadır. Bu da ajanın içinde bulunduğu çevreyi keşfetmesine imkân sağlamaktadır. Ajan daha fazla eylem gerçekleştirdikçe epsilon değeri düşmekte ve ajan bildiği mevcut iyi eylemleri denemeye başlamaktadır. Epsilon değeri başlangıçta 1 olup seçilen eylemler rastgele yapılmaktadır. Ajan, eğitim aşamasının sonuna doğru daha az keşif daha fazla sömürü yaklaşımını kullanır.

Bellman denklemi ajanın her bir adımından sonra Q -tablosunun nasıl güncelleneceğini belirtir. Bu denklemi özetlemek gerekirse ajan algılamış olduğu o anki değerini bilinen en iyi eylemine ait tahmini gelecekteki en iyi ödül ile günceller ve

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a_{t+1})) \quad (3.35)$$

gibi olur. Tüm olası durum-eylem çiftlerinin Q değerlerini kapsamlı bir şekilde hesaplamak maliyetli olacaktır. Q değerlerini bu şekilde hesaplamak yerine, sinir ağı gibi herhangi bir fonksiyon tahmincisi kullanarak yaklaşık değerler hesaplanabilir. Q fonksiyonunu bir θ parametresi ile parametreleştirebilir ve θ parametresinin sadece sinir ağının parametresi olduğu yerde Q değeri hesaplanabilir. Yani, sadece ortamın durumuyla sinir ağı beslenir ve o durumdaki tüm olası eylemlerin Q değeri böylece hesaplanır. Q değerlerini elde ettikten sonra, maksimum Q değerine sahip olan en iyi eylem seçimi yapılır. Q değerini tahmin etmek için bir sinir ağı kullandığında, sinir ağına Q ağı denir ve Q değerini yaklaşık olarak belirlemek için bir DSA kullanıldığında ise DSA'ya DQA denir.

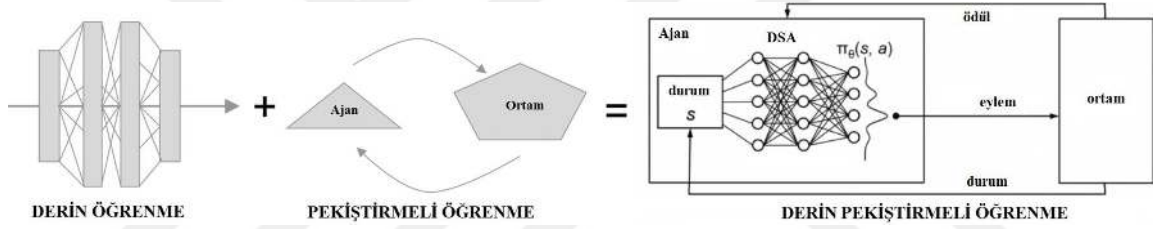
3.5.5. Derin Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme problemleri için yapay sinir ağlarının kullanımı 1990'larda başlamıştır. Klasiklerden çalışmalardan biri, Gerald Tesauro ve diğerleri tarafından oluşturulan tavla oynayan bilgisayar programı TD-Gammon'dur [248]. TD-Gammon, pekiştirmeli öğrenme aracılığıyla masa pozisyonlarını kendi başına değerlendirmeyi öğrenerek tavla oynamayı öğrenmiştir. Uygulanan teknikler tam olarak DPÖ olarak kabul edilmese de, TD-Gammon,

karmaşık pekiştirmeli öğrenme sorunlarını çözmek için sinir ağlarını kullanan, ilk başarılı çalışmalardan biridir.

DPÖ birçok alanda kullanılmaktadır. Atari oyunları oynama, Mnih tarafından geliştirilen klasik oyun oynayan ajanlara verilebilecek en basit örneklerden biridir [249]. Mnih ve arkadaşları yapmış oldukları çalışmada KSA ile pekiştirmeli öğrenmeyi birleştirerek yeni bir yaklaşım sunmuştur. Bu çalışma, DPÖ alanında yeni bir süreci başlatmıştır.

Derin öğrenme ile pekiştirmeli öğrenmenin birleştirilmesi, gerçek hayatta karşılaşılan birçok problemi çözmemize yardımcı olabilme potansiyeline sahiptir. Şekil 3.33'te birleştirilen iki öğrenmenin ortaya çıktığı yapı örneği görülmektedir. DSA'lar pekiştirmeli öğrenmede değer ve politika fonksiyonlarında fonksiyon uydurma için kullanılır. Pekiştirmeli öğrenme ve DSA'ları birleştirmenin temel motivasyonlarından biri, PÖ'de ortaya çıkan yüksek boyutlu durumlar uzayını derin YSA'ların ihtiva edebilme potansiyelidir. DPÖ'nün ayırt edici özelliği deneme yanılma yoluyla eş zamanlı olarak sıralı, değerlendirici ve doğrusal olmayan güçlü fonksiyon uydurucuların kullanılarak örneklediği bir geri besleme ile öğrenmenin sağlanmasıdır.



Şekil 3.33. Derin öğrenme ile pekiştirmeli öğrenmenin birleştirilmesi

Pekiştirmeli öğrenme algoritmalarından Q-öğrenmede değer fonksiyonu

$$Q(s, a) = r(s) + \gamma \max_{a'} \sum_{s'} P(s'|s, a) Q(s', a') \quad (3.36)$$

olarak tanımlanmaktadır. Denklem 3.36 olasılıksal ortamlar için Bellman denklemi olarak bilinir. Bu denkleme göre durum eylem çiftinin Q-değeri o anki ödül toplamına ve sonraki Q-değerlerinin toplamına eşittir. Denklem 3.36 ayrıca ayrık bir özelliğe sahiptir. Diğer taraftan gerçek hayattaki uygulamalarda durumlar ve eylemler süreklilik özelliğine sahiptir. Bu sebeple değer fonksiyonu için etkin bir fonksiyon uydurma yöntemi gerekir. Burada sinir ağları bu ihtiyacı karşılamak için devreye girmektedir. Değer fonksiyonundaki her bir durum ve Q-değeri sinir ağlarındaki gizli katmanlar kullanılarak hesaplanır. Sinir ağları geriye yayılım algoritması kullanılarak eğitilir. DPÖ ile yapay zekâda yeni gelişmeleri başlatan Mnih ve arkadaşlarının yaptığı çalışmada derin öğrenme, pekiştirmeli öğrenmede kullanılarak DPÖ ortaya çıkmış ve bu alandaki yeni çalışmaların önünü açmıştır [249]. Pekiştirmeli öğrenmede daha sonra alınan ödüllerin etkisi hemen alınan ödüllere nazaran daha azdır. DSA'lar ile Q-fonksiyonları ajanın bir sonraki eylemi seçerken uzun vadedeki ödülleri dikkate almasını sağlar. DSA'ların getirmiş olduğu bir başka fayda ise ajan ile ortamın

arasında ihtiyaç duyulan etkileşim sayısının azaltılmasıdır. Bu da performansın artmasına ve algoritmadaki verinin verimini arttırmaktadır.

3.5.6. DPÖ Yöntemleri

DPÖ’de politika, değer ve model tabanlı olmak üzere üç temel yaklaşım vardır. Ayrıca bu yöntemlerin politika-değer ya da değer-modelin birlikte kullanıldığı karma yaklaşımlarda bulunmaktadır.

Politika tabanlı algoritmalarda amaç π ile gösterilen politikayı öğrenmektir. Bu algoritmalar sezgisel olup ajanın bir ortamda bazı eylemleri yerine getirmesi durumunda politika öğrenmesi anlamlı olmaktadır. Belirli bir andaki durumda iyi bir eylemden bahsedebilmek için politika fonksiyonu olan π ’nin s durumunu giriş olarak alıp a aksiyonunu üretmesi gerekir. Bu da ajanın farklı koşullarda iyi kararlar alabildiğini göstermektedir. REINFORCE algoritması en bilinen politika tabanlı bir algoritmadır. Politika tabanlı algoritmaların en önemli avantajı optimizasyon yöntemlerinin genel bir sınıfı olmalarıdır. Ayırık ya da sürekli türden eylemlerden oluşan problemlere uygulanabilirler. İyi tasarlanmış politikalar ajanın hedefini maksimize eden τ yörüngesini üreten eylemler oluşturmalıdır. Ajanın hedefi

$$J(\tau) = E_{\tau \sim \pi} [\sum_{t=0}^T \gamma^t r_t] \quad (3.37)$$

ile hesaplanır. Değer tabanlı algoritmalarda ajan $V^\pi(s)$ veya $Q^\pi(s,a)$ ’yı öğrenmeyi amaçlamaktadır. Öğrenilen değer fonksiyonu, (s,a) çiftini ölçmek ve bir politika oluşturmak için kullanır. Örneğin bir ajanın politikası, s durumunda en yüksek değerli hesaplanan $Q^\pi(s,a)$ ’ya sahip a eyleminin seçilmesi olabilmektedir. Saf değer tabanlı yaklaşımlar için bir politikaya dönüştürülmesi daha kolay olması hasebiyle $Q^\pi(s,a)$ ’nın öğrenilmesi $V^\pi(s)$ ’ya göre daha yaygındır. Bunun sebebi $Q^\pi(s,a)$ ’ya bakıldığında hem durum hem de eylem bilgisi içerirken $V^\pi(s)$ ’da sadece durum bilgisi bulunmaktadır. Değer tabanlı algoritma türlerine bakıldığında SARSA bilinen en temel algoritmalarından biridir. Ancak eğitim aşamasındaki yüksek tutarsızlığı ve etkileşim verimsizliği sebebiyle pek tercih edilmemektedir. Derin Q-Öğrenme (DQÖ) ve aynı nesilden gelen çift DQA, PER gibi yaklaşımlar daha popüler değer tabanlı algoritma türlerine örnek verilebilir.

Değer tabanlı algoritmaların politika tabanlı algoritmalarla göre etkileşim verimliliği daha yüksektir. Bunun sebebi düşük tutarsızlığa sahip olmaları ve ortamdan toplanan verinin daha verimli kullanılmasıdır. Ancak bu durum bu algoritmaların her zaman optimum noktaya yakınsayacağı anlamına gelmemektedir. Değer tabanlı algoritmalarındaki kısıtlamanın, standart formülasyonunda ayırık eylem uzaylı ortamlara uygulanabilir olduğu görülmektedir. Sürekli eylem uzaylarına da uygulanabilmesi için QT-OPT algoritması gibi son zamanlarda bazı ilerlemeler kaydedilmiştir.

Model tabanlı algoritmalarla bakıldığında bir modele ait ortamın geiş dinamiklerini ğrenmekte ya da bilinen bir dinamik modelden faydalanmaktadır. Bir ajan ortamın $P(s, a | s')$ gibi bir modeline sahipse yrnge bilgisini belirli zamana kadar tahmin ederek gelecekte ne olacađına dair ngrde bulunabilmektedir. Ortam bir s durumunda ise, ajan bir dizi eylem gerekleřtirerek $P(s, a | s')$ 'ye gre s durumunun nasıl deđiřebileceđini tahmin edebilmektedir. Bylece tahmin edilen yrnge ajanın bir model kullanarak meydana gelmektedir. Bir ajan farklı eylem dizileriyle farklı yrnge tahminlerinde bulunabilir. Daha sonra bu seeneklere bađlı olarak en iyi eylem seiminde bulunabilmektedir. Model tabanlı algoritmalar daha ziyade bir hedef durumuna sahip oyunlara uygulanmaktadır. Satranta olduđu gibi oyunu kaybetme ya da kazanma durumu bir hedef durumu olarak belirlenebilmektedir. Bunun sebebi geiş fonksiyonlarının hibir dl modelleyememesidir. Eylemleri planlamada kullanabilmek iin ajanın hedefi ile ilgili bazı bilgilerin her bir durumda kodlanması gerekmektedir.

Monte Carlo Ađa Araması (MCAA) algoritması geiş fonksiyonları bilinen deterministik ayrık durum uzaylara sahip problemlere uygulanabilen en bilinen model tabanlı algoritmalarından biridir. Satran ve Go gibi masa oyunları bu kategoriye girmektedir. Son zamanlara kadar MCAA birok Go oyununa katkı sađlamıřtır [123,124]. MCAA herhangi bir makine ğrenmesi algoritmasını kullanmayıp bir oyunun durumlarını ve bu durumların deđerlerini keřfetmek iin eylem dizilerini rastgele rneklemektedir. Diđer yntemler geiş dinamiklerini ğrenme ile ilgilidir. Dinamiklerin ğrenilebilmesi iin ajanın bir ortamda gerek geiş rneklerini (s, a, r, s') toplamak iin eylem gerekleřtirmesi gerekmektedir. Model tabanlı algoritmalarda iyi bir model ngrye sahip bir ajanın oluřturulmasını sađlaması sebebiyle olduka caziptir. Bu modeller bazı senaryolar oluřturup eylemlerinin sonularını gerek anlamda bir ortamda hareket etmeden yorumlayabilirler. Bu da robotik gibi ortamdan veri edinmede iřlem maliyetinin yksek olduđu ve zaman gerektiren durumlarda nemli bir avantaj sađlayabilir. Politika ve deđer tabanlı yntemlere nazaran model tabanlı algoritmalar iyi politikaların ğrenilmesi iin daha az veri rneđine ihtiya duyulmaktadır. Bunun sebebi bir modelin olması ajanın gerek tecrbelerine hayali tecrbelerini de ilave etmesidir. Bununla birlikte model tabanlı algoritmalarda modelin edinilmesi olduka zordur. ođu ortam olasılıksal olup geiş dinamikleri bilinmemektedir. Bu tr durumlarda modelin ğrenilmesi gerekmektedir. Bu yaklařım yeni olup bir takım zorluklarla karřılařılmaktadır. Bu zorluklardan ilki zellikle geiřlerin ziyadesiyle karmařık olduđu geniř bir durum uzayına ve eylem uzayına sahip bir ortamın modellenmesidir. Diđer bir zorluk ise modeller sadece ortam geiřlerini gelecekteki adımlarda dođru bir řekilde tahmin ederken faydalıdır. Modelin dođruluđuna bađlı olarak tahmin hatalarının her bir adımda eklenmesi modelin gvenilirliđini olumsuz ynde etkilemektedir. Model tabanlı yaklařımların uygulanmasında iyi modellerin olmaması nemli bir kısıtlama olarak bilinmektedir.

Model tabanlı ve modelsiz algoritmalar arasındaki en önemli ayrıma bağlı olarak pekiştirmeli öğrenme algoritmaları sınıflandırılabilir. Model tabanlı bir algoritma önceden bilinmesi halinde basitçe ortamın geçiş dinamiklerinden faydalanan herhangi bir algoritmadır. Modelsiz algoritmalar ise ortam geçiş dinamiklerini faydalanmayan algoritmalar. Modelsiz algoritmalar ise ortam geçiş dinamiklerini faydalanmayan algoritmalar.

Karma yaklaşımlar ise şuna kadar bahsedilen yaklaşımlardan iki ya da daha fazla yaklaşımın bir araya getirilmesiyle geliştirilen algoritmalar. Yaygın olarak kullanılan algoritmalar bir grubu bir politika ve değer fonksiyonunu öğrenmektedir. Bu algoritmalar aktör-kritik algoritmaları olarak bilinen algoritmalar. Aktör geliştirdiği politikalara göre bazı eylemler gerçekleştirmektedir. Kritik ise değer fonksiyonu ile yapılan eylemlerin kritiğini yapmaktadır. Eğitim aşaması boyunca öğrenilen bir değer fonksiyonu bir politikaya ortamda mevcut olan ödül dizisinden daha fazla bilgi içeren bir geri besleme sinyali sağlayabilmektedir. Politika öğrenilen değer fonksiyonu tarafından sağlanan bilgiyi kullanarak öğrenmektedir. Politika daha sonra politika tabanlı algoritmalar olduğu gibi eylemler üretmektedir. Aktör-kritik algoritmalar aktif bir araştırma alanı olup bu alanda son yıllarda Trust Region Policy Optimization (TRPO) [250], Proximal Policy Optimization (PPO) [251], Deep Deterministic Policy Gradients (DDPG) [252], ve Soft Actor-Critic (SAC) [253] gibi algoritmaların geliştirilmesine yol açmıştır.

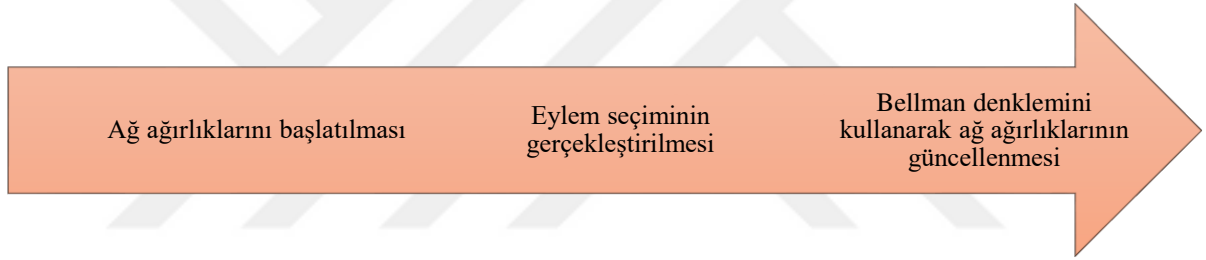
Karma yaklaşımlar bir değer fonksiyonu veya politika ile birlikte ortam geçiş dinamiklerinin modelini de kullanabilirler. 2016 yılında DeepMind şirketindeki araştırmacılar AlphaGo'yu geliştirdiler [153]. AlphaGo'da MCAA, V^π değer fonksiyonu ve π politikasıyla birleştirilmiştir.

DPÖ algoritmalarının önemli bir ayırt edici özelliği de politikalı ya da politikasız olmalarıdır. Bu durum eğitim aşamasında veriden nasıl faydalandıklarına göre farklılık göstermektedir. Politikalı öğrenme de eğitim aşamasında sadece anlık politikadan üretilen veriden faydalanılır. Eğitim aşaması devam ettiği sürece her bir iterasyonda eğitim verisini üretmek için anlık politika kullanılır. Sonuç olarak eğitim aşamasından sonra tüm veri kullanılamaz duruma geldiği için atılmaktadır. Bu da politikalı öğrenme yöntemlerinde etkileşim verimsizliğine neden olmaktadır. Bir başka deyişle daha çok veriye ihtiyaç duymaktadır. Politikalı öğrenme yöntemlerine REINFORCE, SARSA, Actor-Critic ve PPO algoritmaları örnek olarak verilebilir. Politikasız öğrenme de ise politikalı öğrenmedeki gibi bir gereklilik yoktur. Daha önce elde edilen veriler eğitim aşamasında tekrar kullanılabilir. Sonuç olarak politikasız öğrenme yöntemleri etkileşim verimliliğine sahiptir. Ancak bu durum daha fazla bellek ihtiyacı doğurmaktadır. Politikasız öğrenme yöntemlerine DQA ve devamı niteliğindeki algoritmalar örnek olarak verilebilir.

Derin Q-Öğrenme Algoritması

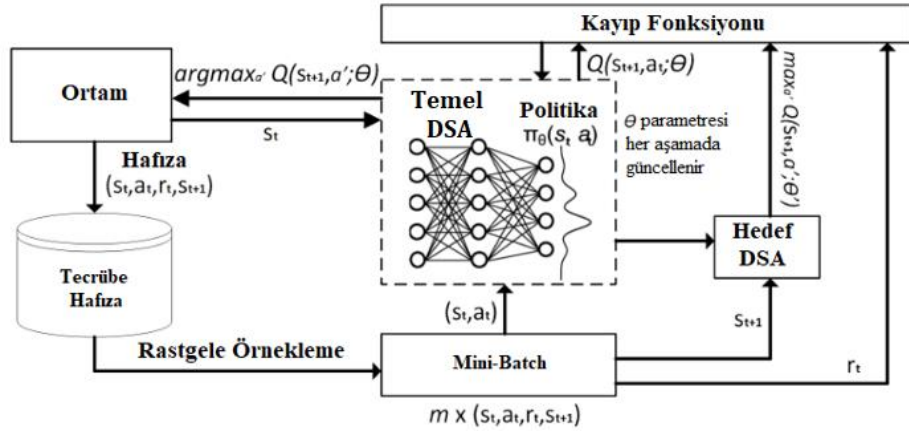
DQÖ algoritması, DeepMind tarafından 2015 yılında geliştirilmiştir [254]. Pek çok Atari oyununda pekiştirmeli öğrenmeyi ve derin sinir ağlarını birleştirerek insanüstü düzeyde performanslar göstermiştir. DQÖ algoritması, Q-Öğrenme adlı klasik bir PÖ algoritmasını DSA ve *tecrübe hafızası-deneyim tekrarı/tekrar arabelleği* (*Experience Replay-Replay Buffer*) adı verilen bir tekniğin geliştirilmesiyle öne çıkan değer tabanlı bir yaklaşımdır. Ajanın deneyimi ile ilgili bilgi edinmek için *tecrübe hafızası* adı verilen bir arabellek kullanılır ve bu deneyime dayanarak DSA eğitilir.

DQÖ algoritmasında ilk olarak temel ve hedef ağlara ait ağırlıklar başlatılır. Ardından epsilon-greedy keşif stratejisine göre eylem seçimi yapılır. Son olarak Bellman denklemi yardımıyla ağ ağırlıkları güncellenir. Q-öğrenme ile DQÖ arasındaki fark, Q-tablosunun nasıl uygulandığıdır. DQÖ’de Q-tablosunun yerini sinir ağları almaktadır. Bir durum-eylem çiftini Q-değeri ile eşlemek yerine sinir ağları giriş olarak alınan durumları eylem-Q değeri çifti ile eşleştirmektedir. Şekil 3.34’te DQÖ süreci görülmektedir.



Şekil 3.34. DQÖ algoritması süreci

DQÖ’de öğrenme süreci için *temel* (*Q*) ve *hedef* (*target*) ağ olmak üzere iki sinir ağı kullanılmaktadır. Bu iki sinir ağı aynı mimariye sahip olup farklı ağırlıklardan oluşmaktadır. Her adımda temel ağdan alınan ağırlıklar hedef ağına kopyalanmaktadır. Bu iki ağın birlikte kullanılması, öğrenme sürecinin istikrarlı bir şekilde sürdürülmesi ve algoritmanın daha etkin bir şekilde öğrenmesine katkı sağlamaktadır. Temel ve hedef ağın her ikisi de giriş katmanından alınan durumları çıkış katmanındaki eylemlerle eşleştirmektedir. Çıkıştaki eylemler modelin tahmin edilen Q-değerlerini temsil etmektedir. Tahmin edilen en yüksek Q-değerine sahip eylem o durumdaki bilinen en iyi eylemdir. Bir eylem seçimi yapıldıktan sonra ajan bu eylemi gerçekleştirir. Ardından temel ve hedef ağlar Bellman denklemine göre güncellenir. DQÖ yönteminde *tecrübe hafızası* tekniği kullanılarak ajan ortam ile ilgili bilgi edindikten sonra temel ve hedef ağları günceller. Tecrübe hafızası tekniği durum, eylem, ödül ve sonraki durum parametrelerinin hafızada tutularak öğrenme sürecinin gerçekleştirilmesini sağlamaktadır. Şekil 3.35’te DQÖ yöntemine ait bir yapı görülmektedir.



Şekil 3.35. DQÖ yöntemine ait bir gösterim [255]

Hafızadan rastgele mini-yığın boyutunda deneyim örneklenerek DQA eğitilir. Ancak, burada eğitimin tam olarak nasıl olduğu ve DQA'nın, bu mini-yığın boyutundaki deneyimi kullanarak optimal Q fonksiyonunu uydurmayı nasıl öğrendiğine bakılmalıdır. Kayıp fonksiyonu (loss function) hesabı yapılarak öğrenme durumu değerlendirilebilir.

DQA'da amaç, sadece sürekli bir değer olan Q değerini tahmin etmektir. Böylece, DQA'da temel olarak bir regresyon görevi gerçekleştirilir. Genellikle regresyon görevi için *kayıp fonksiyonu* olarak ortalama karesel hatayı (OKH) kullanılır. OKH

$$MSE = \frac{1}{K} \sum_{i=1}^K (y_i - \hat{y}_i)^2 \quad (3.38)$$

ile hesaplanır. Burada y hedef değer olup, $\hat{y}$ tahmin edilen değerdir ve K , eğitim örneklerinin sayısıdır. Hedef Q değeri ile tahmin edilen Q değeri arasındaki OKH'yı minimize ederek ağı eğitilir. Optimum Q değeri ile tahmin edilen Q değeri arasındaki hatayı en aza indirerek ağı eğitilebilmesi için hedef Q değeri *en uygun (optimal) Q değeri* olmalıdır. Optimal Q değeri, Bellman optimalite (optimality) denklemi kullanılarak

$$Q^*(s, a) = r + \gamma \max_{a'} Q^*(s', a') \quad (3.39)$$

ile elde edilebilir. Bellman optimalite denkleminde göre, optimal Q değeri, bir sonraki durum-eylem çiftinin sadece ödülün ve indirgenmiş maksimum Q değerinin toplamıdır. Dolayısıyla kayıp, hedef değer (optimum Q değeri) ile tahmin edilen değer (DQA tarafından tahmin edilen Q değeri) arasındaki fark olarak tanımlanabilir ve kayıp fonksiyonunu L

$$L(\theta) = Q^*(s, a) - Q_\theta(s, a) \quad (3.40)$$

gibi ifade edilir. Burada, Denklem 3.39'daki ifade kullanılıp değiştirilerek

$$L(\theta) = r + \gamma \max_{a'} Q(s', a') - Q_\theta(s, a) \quad (3.41)$$

şeklinde yazılabilir. Tahmin edilen Q değerine benzer şekilde, θ parametrelili aynı DQA'yı kullanılarak hedefteki (target Q value) sonraki durum-eylem çiftinin Q değeri hesaplanabilir. Böylece kayıp fonksiyonu

$$L(\theta) = r + \gamma \max_{a'} Q_{\theta}(s', a') - Q_{\theta}(s, a) \quad (3.42)$$

gibi yeniden yazılabilir. İlgili denklemlerde gösterildiği gibi, hem hedef değer hem de tahmin edilen Q değeri θ ile parametrelendirilir. Kaybı, yalnızca hedef Q değeri ile tahmin edilen Q değeri arasındaki fark olarak hesaplamak yerine, kayıp fonksiyonu olarak OKH kullanılır. Ajanın deneyimi, *tecrübe hafızası* adı verilen bir arabellekte depolanır. Bu nedenle, tecrübe hafızasından K sayısı kadar geçişli rastgele mini-yığın (s, a, r, s') örneklenir ve OKH en aza indirgenerek ağ eğitilir. Böylece kayıp fonksiyonu

$$L(\theta) = \frac{1}{K} \sum_{i=1}^K (r_i + \gamma \max_{a'} Q_{\theta}(s'_i, a') - Q_{\theta}(s_i, a_i))^2 \quad (3.43)$$

ile ifade edilebilir. Kayıp fonksiyonu minimize edilerek DQA eğitilir. Optimum θ parametresini bulunarak kayıp fonksiyonu minimize edilir. Bu nedenle, optimal θ parametresini bulmak için gradyan inişini kullanılır. $\nabla_{\theta} L(\theta)$ kayıp fonksiyonu gradyanını hesaplanır ve θ DQA parametresi

$$\theta = \theta - \alpha \nabla_{\theta} L(\theta) \quad (3.44)$$

ile güncellenir. *Hedef ağ*, bir sonraki durum-eylem çiftinin sadece ödülün ve indirgenmiş maksimum Q değerinin toplamını bulur. Hedef ve tahmin edilen Q değerlerinde bir sonraki durum-eylem çiftinin Q değeri, θ tarafından parametrelendirilen aynı ağ kullanılarak hesaplanır. Buradaki sorun, hedef ve tahmin edilen değerlerin aynı θ parametresine bağlı olması nedeniyle, OKH'de kararsızlığa neden olması ve ağdaki öğrenmenin yetersiz kalmasıdır. Ayrıca, bu durum eğitim sırasında çok fazla sapmaya neden olur.

θ güncellendiğinde hem hedef hem de tahmin edilen değerler değişir. Böylece, tahmin edilen değer hedef değerle aynı olmaya çalışılır, ancak θ ağ parametresindeki güncelleme nedeniyle hedef değer değişmeye devam eder. Bunu önlemek için, hedef değeri bir süreliğine dondurabilir ve yalnızca tahmin edilen değer hesaplanarak tahmin değerinin hedef değerle eşleşmesi sağlanabilir. Bunu yapmak için, hedefteki bir sonraki durum-eylem çiftinin Q değerini hesaplamak için hedef ağ adı verilen başka bir sinir ağ kullanılır. Hedef ağın parametresi θ' ile temsil edilir. Böylece, Q değerlerini tahmin etmek için kullanılan ana DQA, gradyan inişini kullanarak optimal parametre θ' öğrenir. Hedef ağ bir süre dondurulur ve ardından hedef ağ parametresi θ' , yalnızca ana DQA parametresi θ kopyalanarak güncellenir. Hedef ağın bir süre dondurulması ve ardından θ' parametresinin ana ağ parametresi θ ile güncellenmesi eğitimi stabilize eder. Dolayısıyla, kayıp fonksiyonu

$$L(\theta) = \frac{1}{K} \sum_{i=1}^K (r_i + \max_a Q_{\theta'}(s'_i, a') - Q_{\theta}(s_i, a_i))^2 \quad (3.45)$$

ile yeniden yazılabilir. Hedefteki sonraki durum-eylem çiftinin Q değeri, θ' ile parametrelenen hedef ağ tarafından hesaplanır ve tahmin edilen Q değeri, θ ile parametrelenen ana DQA tarafından hesaplanır.

DQA algoritması aşağıdaki adımlarda verilmektedir:

1. Ana DQA parametresi θ' 'yi rastgele değerlerle başlat
2. Ana ağ parametresi θ' 'yi kopyalayarak hedef ağ parametresi θ' başlat
3. Tecrübe hafızasını başlat
4. N sayıda bölüm (episode) için 5. adımı gerçekleştir
5. Bölümdeki her adım için, yani $t = 0$ için. . . , $T-1$:
 - 1) s durumunu gözlemler ve epsilon-greedy politikasını kullanarak bir eylem seçimi yap, yani ϵ değeri ile rastgele a eylemini seç ve $1-\epsilon$ ile $a = \arg\max_a Q_{\theta}(s, a)$ eylemini seç
 - 2) Seçilen eylemi gerçekleştir, sonraki s' durumuna geç ve r ödülünü al
 - 3) Geçiş (transition) bilgilerini tecrübe hafızasında sakla
 - 4) Tecrübe hafızasından K geçişlerini mini-yığın boyutunda rastgele örnekle
 - 5) Hedef değeri hesapla
 - 6) Kayıp değerini hesapla
 - 7) Kayıp fonksiyonunun gradyanlarını hesapla ve gradyan inişini kullanarak ana ağ parametresi θ' 'yi güncelle
 - 8) Hedef ağ parametresi θ'' 'nü birkaç zaman adımı için dondur ve daha sonra sadece ana ağ parametresi θ' 'yi kopyalayarak güncelle

Çift DQA Algoritması

Çift DQA Algoritması, Hado van Hasselt ve arkadaşları tarafından 2016 yılında önerilmiştir [256]. DQA ile ilgili sorunlardan biri, hedefteki bir sonraki durum-eylem çiftinin Q değerini olduğundan fazla tahmin etme eğiliminde olmasıdır. Olduğundan fazla tahmin etme eğilimi max operatörünün varlığından kaynaklanmaktadır. Çift DQA Algoritması ile bu sorunun gözlenen aşırı tahminlerinin azaltılması amaçlanmaktadır.

$$y = r + \gamma \max_{a'} Q_{\theta'}(s', a') \quad (3.46)$$

Denklem 3.46'daki hedef değer hesaplaması Denklem 3.47'deki gibi değiştirilerek fazla tahmin sorunu çözülebilir.

$$y = r + \gamma Q_{\theta'}(s', \arg\max_{a'} Q_{\theta}(s', a')) \quad (3.47)$$

Denklem 3.47’de görüldüğü gibi, hedef değerin hesaplanmasında iki Q fonksiyonu var. Ana ağ parametresi θ tarafından parametrelenen bir Q fonksiyonu, eylem seçimi için kullanılır ve hedef ağ parametresi θ' tarafından parametrelenen diğer Q fonksiyonu ise Q değeri hesaplaması için kullanılır. Denklem 3.47 iki ayrı adımda incelenebilir:

Eylem seçimi: İlk olarak, θ tarafından parametrelenen ana ağ kullanarak sonraki tüm durum eylem çiftlerinin Q değerleri hesaplanır ve ardından maksimum Q değerine sahip olan a' eylemi seçilir.

Q değeri hesaplaması: a' eylemini seçtikten sonra, seçilen eylem a' için θ' tarafından parametrelendirilen hedef ağ kullanılarak Q değeri hesaplanır.

Hedef değeri hesaplama dışında, çift DQA, DQA ile tamamen aynı şekilde çalışır. Çift DQA algoritmasının DQA algoritmasından farkı, 5.adımda 5.işlemden hesaplanan hedef değerinin yapılmasıdır.

Düello DQA Algoritması

Pekiştirmeli öğrenmedeki en önemli fonksiyonlardan biri, avantaj fonksiyonu olarak bilinir. Avantaj fonksiyonu, Q fonksiyonu ile değer fonksiyonu arasındaki fark olarak tanımlanır ve ifade

$$A(s, a) = Q(s, a) - V(s) \quad (3.48)$$

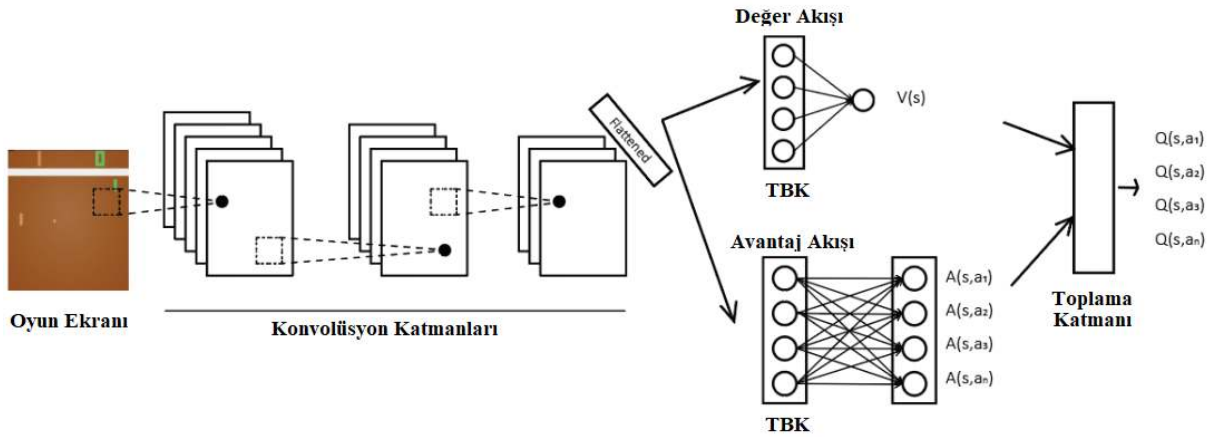
gibi olur. Q fonksiyonu, bir ajanın s durumundan başlayarak a eylemini gerçekleştirerek ve π politikasını izleyerek elde edeceği beklenen toplam ödülü verir. Değer fonksiyonu, bir ajanın s durumundan başlayarak ve π politikasını izleyerek elde edeceği beklenen toplam ödülü verir. Q fonksiyonu ile değer fonksiyonu arasındaki fark, Q fonksiyonunun bize durum-eylem çiftinin değerini vermesi, değer fonksiyonunun ise eylemden bağımsız olarak bir durumun değerini vermesidir. Bir başka deyişle, Q fonksiyonu ile değer fonksiyonu arasındaki fark, bize a eyleminin s durumundaki ortalama eylemlerle karşılaştırıldığında ne kadar iyi olduğunu söyler. Bu nedenle, avantaj fonksiyonu s durumunda, a eyleminin ortalama eylemlere kıyasla ne kadar iyi olduğunu gösterir.

Bir DQA’da durum girdi olarak beslenir ve ağın bu durumdaki tüm eylemleri için Q değerini hesaplar. Q değerlerini bu şekilde hesaplamak yerine, avantaj fonksiyonunu kullanarak Q değerleri hesaplanabilir. Denklem 3.48

$$Q(s, a) = V(s) + A(s, a) \quad (3.49)$$

olarak yeniden yazılabilir. Burada, sadece değer fonksiyonunu ve avantaj fonksiyonunu toplayarak Q değeri hesaplanabilir. Bir ajan için tüm durumlar önemli değildir. Hangi eylem yapılırsa yapılsın her zaman kötü bir ödül veren bir durum olabilir. Bu durumda, durumun her zaman kötü bir ödül vereceği bilirse, durumdaki tüm olası eylemlerin Q değerini hesaplamak yararlı değildir.

Böylece, bunu çözmek için Q fonksiyonu değer fonksiyonu ile avantaj fonksiyonunun toplamı olarak hesaplanabilir. Bu amaçla Zu Wang ve ekibi tarafından 2016 yılında Düello DQA Algoritması önerilmiştir [257]. Değer fonksiyonu ile bir durumdaki tüm eylemlerin değerlerini hesaplamadan bu durumun değerli olup olmadığı anlaşılabilir. Avantaj fonksiyonu ile bir eylemin gerçekten iyi olup olmadığını veya diğer tüm eylemlerle aynı değeri verdiği anlaşılabilir. Q değeri, durum değeri ve avantaj değeri birlikte toplanarak hesaplanır, dolayısıyla değer akışı ve avantaj akışı, toplama katmanı adı verilen başka bir katman kullanılarak birleştirilir ve Q değeri Şekil 3.36’da gösterildiği gibi hesaplanır. Böylece, değer akışı durum değerini hesaplar, avantaj akışı avantaj değerini hesaplar ve toplama katmanı bu akışları toplar ve Q değerini hesaplar.



Şekil 3.36. Toplama katmanı içeren bir düello DQA mimarisi [257]

Ancak burada toplam katmandaki durum değerini ve avantaj değerini toplamak ve Q değerini hesaplamak bir tanımlanabilirlik (identifiability) sorununa yol açabilir. Bu problemin çözülmesi için, avantaj fonksiyonu seçilen eylem için sıfır avantaja sahip olacak şekilde oluşturulur. Bu da, Denklem 3.50’de gösterildiği gibi, eylem uzayındaki tüm eylemlerin ortalama avantaj değerini çıkarılarak çözülebilir.

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a') \right) \quad (3.50)$$

$\mathcal{A}$, eylem alanının uzunluğunu belirtir. Böylece, Q değerini parametrelerle hesaplamak için Denklem 3.50’nin son hali, Denklem 3.51’deki gibi yazılabilir.

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta, \alpha) \right) \quad (3.51)$$

Denklem 3.51’de, θ konvolüsyon ağının parametresidir. β , değer akışının parametresi ve α , avantaj akışının parametresidir. Q değerini hesapladıktan sonra eylem seçimi

$$a = \arg \max_a Q(s, a; \theta, \alpha, \beta) \quad (3.52)$$

ile yapılabilir. Özet olarak, düello DQA ile standart DQA arasındaki fark, düello yapan bir DQA'da Q değerlerini doğrudan hesaplamak yerine, durum değeri ile avantaj değeri birleştirilerek hesaplanır.



4. UYGULAMALAR

Bu bölümde, uygulamalara yer verilecektir. Uygulamalarda önce, yEMG sinyalleri kullanılarak kavrama sinyallerinin sınıflandırılması gerçekleştirilmiştir. Sonra DKQÖ algoritması ile Pong oyunu uygulaması anlatılmıştır. Daha sonra Webots simülatöründe DKQÖ algoritmasına dayalı olarak öğrenen bir robotik kavrama uygulamasının aşamalarına yer verilmiştir.

4.1. yEMG Verileri Kullanılarak 1B-KSA Modeli ile Kavrama Uygulaması

Pratikte kavrama eylemleri insan için basit olmasına karşın bilgisayarlar tarafından sınıflandırma gibi çok çalışılmış bir problemin kavramadaki çözümü pek kolay olmamaktadır. Bu uygulamadaki amaç, kavramanın özellikle robotlara uygulamada ciddi zorluklar olduğunu önceden görmektir.

Kavrama, kontrol gereksinimleri nedeniyle robotik ve protetik uygulamalarda zorlu bir problemidir. Görsel algılama ve ElektroMiyoGrafı (EMG) sinyallerini analiz etme, kavrama yetenekleri için robotlara ve protez amputelere girdi vermenin iki yoludur [258]. EMG, iskelet kaslarının uygunluk durumunu değerlendiren bir tanı yöntemidir. EMG sinyallerinin incelenmesi veya değerlendirilmesi uzmanlar için zaman alıcı ve zahmetlidir. Bu nedenle, EMG kavrama sinyallerinin tespiti ve sınıflandırılmasında doğruluk oranını artırmak için yapay zekâdaki en son yöntemlerin kullanımı öne çıkmaktadır. Son zamanlarda, derin öğrenme mimarileri, sağlık koşullarının teşhisi, bilgisayarla görme ve İnsan Makine Etkileşimi (İME) gibi birçok mühendislik uygulamasında kullanılmaktadır [1,259,260]. Bu çalışmada, altı tip el hareketini sınıflandırmak için yeni bir derin tek boyutlu KSA modeli (1B-KSA) önerilmiştir. yEMG kullanılarak uygulanan 1B-KSA modeli, altı el hareketini sınıflandırmada %94,94 ile en yüksek doğruluğu elde etmiştir. Önerilen modelin öne çıkan tarafı, yalnızca bir kanal verisi kullanarak çeşitli zor kavramaların otomatik sınıflandırmasını gerçekleştirebilmesidir. Geliştirilen model daha fazla veri ile test edilmeye hazır olup ve kas-iskelet sistemi rahatsızlıklarına yardımcı olmak için kullanılabilir.

4.1.1. İlgili Literatür

Biyoelektrik sinyallerin genliklerinin oldukça küçük olması nedeniyle sağlıklı bir şekilde analiz edilebilmeleri için genliklerinin yeterli miktarda kuvvetlendirilmesi gerekmektedir [261]. Biyoelektrik sinyaller insandaki motor becerileri ve fiziksel hareket bilgilerini içermektedir [262]. İnsan vücudunda ElektroKardiyoGrafı (EKG), ElektroEnsefaloGrafı (EEG), ElektroOküloGrafı (EOG) ve EMG gibi farklı biyoelektrik sinyaller bulunmaktadır [263–267]. EMG, sinir hücresi ve kaslardan üretilen elektriksel aktivitenin analizi, değerlendirilmesi ve kaydedilmesini sağlayan bir miyoelektrik aktivitedir. EMG sinyalleri, kaslarla ilgili birçok bilgi içermektedir ve klinik

çalışanlarına doğru teşhis yapma açısından önemli katkılar sağlamaktadır. EMG; biyometri, sağlık hizmetleri, protez yada rehabilitasyon cihazları ve birçok İME uygulamalarında kullanılmaktadır [263,268–271]. Hem robotik hem de protez el insan benzeri fiziksel fonksiyonları yerine getirmek için akıllı robotik endüstrisinde kullanılmaktadır. Bu durum kavrama becerilerinde yeni gelişimlerin önünü açmaktadır. EMG sinyallerinin sınıflandırılması rehabilitasyon çalışmalarına yapabileceği katkılar nedeniyle önem arz etmektedir [272,273]. Robotik el, en önemli robotik uzuvlardan biri olup farklı şekildeki nesnelerin kavranması için temel el hareketlerinin gerçekleştirilmesinde önem arz etmektedir. Kavramanın önceden öğrenilmesi ve farklı tipteki eller için uygulanması büyük avantaj sağlamaktadır.

Her geçen gün yeni gelişmelerin yaşandığı yapay zekâ, hayatımızda önemli bir rol oynamaktadır. Yapay zekâ son yıllarda finans, sürücüsüz araçlar, oyun teorisi, robotik, tıp ve sağlık hizmetleri gibi alanlarda önemli gelişimlerin yaşanmasına yol açmıştır [46,260,274–278]. Yapay zekânın söz konusu alanlarda bu denli önem arz etmesinin nedeni derin öğrenmenin getirmiş olduğu yeni yaklaşımlardır [160,188,259,279]. Derin öğrenmede, ImageNet yarışmasında Krizhevsky ve arkadaşları tarafından önerilen AlexNet modelinin elde etmiş olduğu başarı ile yeni bir dönem başlamıştır [151]. Önerilen model, o ana kadar mevcut olan tüm geleneksel yaklaşımlardan daha iyi sonuç vermiştir. Daha sonrasında sonrasında araştırmacılar bu alana daha çok ilgi göstererek yapay zekânın gelişiminde önemli katkılar sunmaya başlamışlardır. Derin öğrenme karmaşık problemlerde büyük verideki öne çıkan özellikleri keşfederek yüksek doğrulukta çözümler sunmaktadır.

Stokastik olması sebebiyle yEMG sinyallerinin analizi zor bir problem olarak karşımıza çıkmaktadır [280]. Bu zorluklardan bazılarını kastaki yorgunluk ve kasılma yoğunluğundaki değişiklikler, zamana bağlı olarak sinyalin karakteristiğinin değişmesi, ilgili uzvun pozisyon ve oryantasyonundaki değişimler ile kullanılan elektrotların yer değişmesi örnek olarak verilebilir [258,281–285]. Son yıllarda yEMG sinyalleri, akıllı protez uzuvlar için önemli bir kontrol kaynağı olması nedeniyle dikkat çekmeye başlamıştır [286,287]. Sapsanis ve arkadaşları, deneysel ayırıştırma modu tabanlı bir yöntem ile yEMG sinyallerini kullanarak temel el hareketlerini sınıflandırmıştır [288]. Bitzer ve arkadaşları yEMG sinyallerini kullanarak kaydedilen farklı parmak hareketlerini tespit edebilen DVM tabanlı bir sistemi geliştirmişlerdir [289]. Kim ve arkadaşları ise manuel uygulanan kuvvet ile yEMG sinyalleri arasındaki ilişkiyi bulmaya yönelik bir çalışma yapmışlardır [290]. Geliştirmiş oldukları algoritma ile yEMG sinyallerini kullanarak kavrama kuvvetlerini tahmin etmişlerdir. Diğer çalışmalarda, dokunma duyusu ile algılanabilen bir sistemin, manipülatörlerin rahat kontrol edilebilmeleri ve kullanılmaları sürecinde daha iyi bir etkileşim sağladığı üzerinde durulmuştur [291,292]. Kayabaşı ve arkadaşları, YSA ile Aşırı Öğrenme Makinesi (AÖM)'nin yEMG veri kümesi üzerindeki performansını karşılaştırılmıştır [293]. Çalışmalarında, AÖM'nin YSA'ya göre daha hızlı ve doğru sonuç verdiği gösterilmiştir.

Literatürde farklı yaklaşımlarla yEMG sinyalleri kullanılarak farklı el hareketleri sınıflandırılmıştır. Ouyang ve çalışma arkadaşları, dört farklı kavrama hareketini sınıflandırmak için adaptif bir nöro-bulanık etkileşimli sistemi kullanmışlardır [294]. Ju ve arkadaşları, bulanık-gauss modeli ile on farklı el hareketini sınıflandırmışlardır [295]. Kakoty ve Hazarika ise 6 farklı kavrama hareketini DVM tabanlı radyal bir kernel ile sınıflandırabilen bir yaklaşım sunmuşlardır [296]. Bir başka çalışmada, üç farklı el hareketi dalgacık eşikli bir yaklaşımla sınıflandırılmıştır [297]. Akben yapmış olduğu çalışmada, uyumluluk ilişkisine dayanan bir yöntemi öznetelik çıkarımı ve kaskad yapıları sınıflandırıcıyı ise altı farklı kavrama hareketini sınıflandırmak için kullanmıştır [298]. Ruangpaisarn ve Jaiy çalışmalarında radyal temelli fonksiyon, k-en yakın komşular (KYK) algoritması, karar ağacı, sıralı minimal optimizasyon ve naive Bayes gibi farklı sınıflandırıcıları karşılaştırmışlardır [299]. Nishad ve arkadaşları, Kraskov entropisini öznetelik çıkarımı ve KYK algoritmasını ise sınıflandırıcı olarak kullanmışlardır [300].

Bahsedilen çalışmaların çoğunda geleneksel makine öğrenmesi yöntemleri ve çok sayıda yEMG elektroda sahip donanımlar kullanılmıştır. Bu yöntemlerin geliştirilmesi oldukça fazla zaman almaktadır ve ayrıca özneteliklerin seçiminde manuel teknikler kullanılmaktadır. Bu çalışmada sınıflandırma için altı farklı yEMG kavrama sinyallerindeki öznetelikleri tamamen otomatik olarak çıkarabilen yeni bir 1B-KSA algoritma önerilmiştir. Uçtan uca 11-katmanlı 1B-KSA modeli öznetelik çıkarımı ve softmax fonksiyonu ise sınıflandırma için kullanılmıştır. Sınıflandırma çalışması temel kavrama hareketlerini içeren erişime açık bir yEMG veri kümesi kullanılarak yapılmıştır [288,301]. Bu veri kümesinde yEMG verilerinin elde edilmesi için 2 elektrot kullanılmıştır. Bu açıdan önerilen bu çalışmanın, tek bir kanal verisi kullanması sebebiyle yeni ve düşük maliyetli bir robotik-protez el tasarımına önemli katkılar sağlayacağı görülmüştür. Bu çalışmadaki motivasyonlardan biri de derin öğrenme yaklaşımı ile yEMG sinyallerinin sınıflandırma başarımının artırılması ve daha fazla zaman ve uğraş gerektiren öznetelik mühendisliği iş yükünün azaltılması olmuştur.

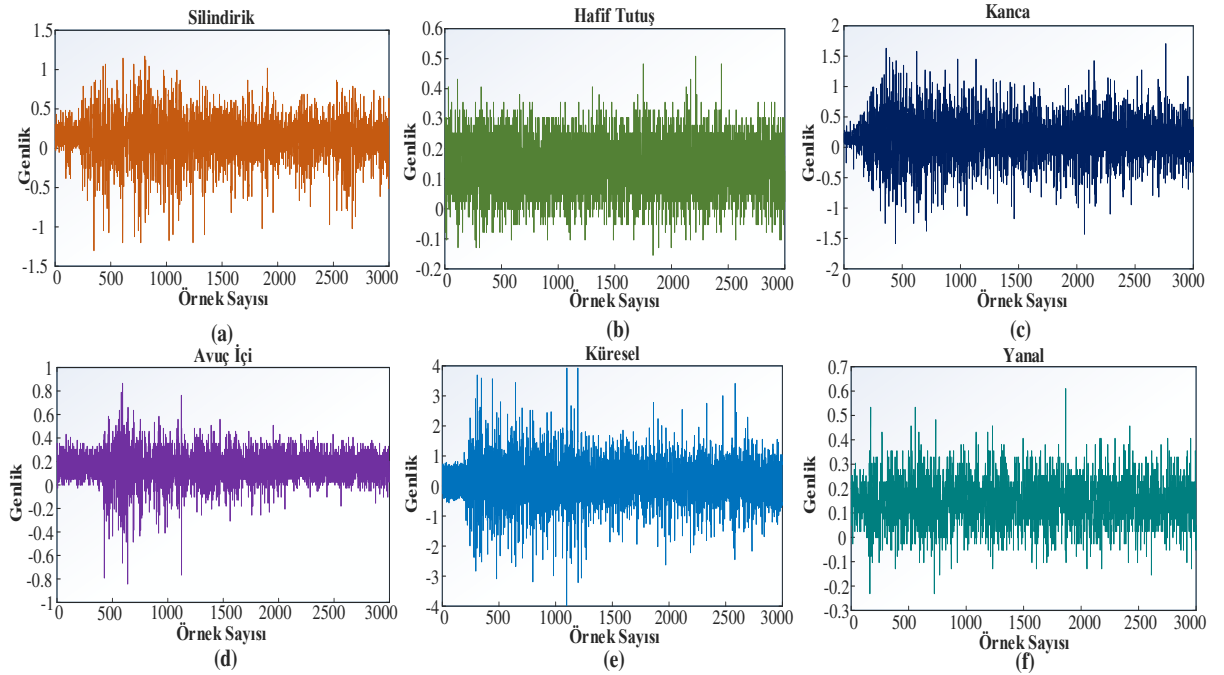
4.1.2. Veri Kümesi

Bu tez çalışmasında, uygulamalar için yEMG sinyallerinden oluşan temel el hareketleri veri seti kullanılmıştır [288,301]. Bu veri seti iki alt veri kümesinden oluşmaktadır. Bu çalışmada bu iki veri kümesine ek olarak her iki veri kümesinin birleştirilmesiyle toplam üç farklı veri kümesi oluşturularak uygulama çalışmaları yapılmıştır. 1. veri kümesi, 2 erkek 3 bayan olmak üzere toplam 5 sağlıklı kişiye ait sinyallerden oluşturulmuştur. 2. veri kümesi ise sağlıklı bir bireye ait müteakip 3 ayrı günde alınan sinyallerden elde edilmiştir. 1. veri kümesi 3000 örnek uzunluğunda iken 2. Veri kümesi 2500 örnekten oluşmaktadır. Veri seti bir sensör sisteminin, kasların bulunduğu belirli noktalara yerleştirilerek bir referans elektrotu ile elastik bantlar yardımıyla oluşturulmuştur. Önkolda iki ayrı noktada Longus ve Brevis ile Flexor Capri Ulnaris ve Extensor Capri Radialis

yEMG elektrotları kullanılmıştır. Burada elde edilen sinyaller iki farklı yEMG sensöründen alındıktan sonra iki kanallı EMG sistemine iletilmiştir. Veri setinin dört önemli özelliği bulunmaktadır. Bunlar; iki yEMG kanalının olması, 500 Hz'lik örnekleme frekansında verinin toplanması, her bireyin altı farklı kavrama şeklini 30'ar defa yapması ve her bir hareketin 6 saniye boyunca kaydedilmesidir. Kavrama sinyalleri sürekli olarak kaydedildikten sonra kavranan nesnenin tipine göre sınıflandırılmıştır. Kavrama tipleri silindir, hafif tutuş, kanca, avuç içi, küresel ve yanal olmak üzere altı farklı tipten oluşturulmuştur. Şekil 4.1'de kavrama tiplerine ait resimler görülmektedir. Tüm kavrama tiplerine ait örnek sinyaller ise Şekil 4.2'de gösterilmiştir.



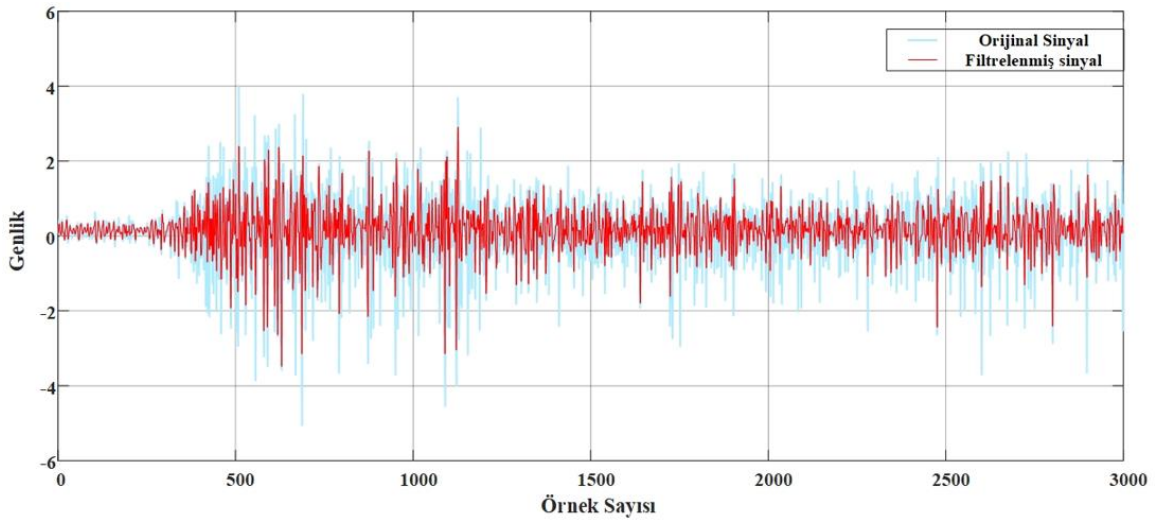
Şekil 4.1. Kavrama tipleri



Şekil 4.2. Tüm kavrama tiplerine ait örnek sinyaller

4.1.3. Önişleme Aşaması

EMG sinyallerinin toplanması için kullanılan elektrotlar ve diğer donanımlardan kaynaklanan gürültüler sebebiyle sinyallerde bir takım bozulmalar meydana gelebilmektedir. Bu durum, sinyallerin sınıflandırılmasını ya da doğru bir teşhis koymayı güçleştirmektedir. Bu çalışmada önişlem aşamasında mevcut gürültüleri elimine etmek için medyan filtre önişlem aşamasında kullanılmıştır. Medyan filtre, doğrusal olmayan bir filtre olup sinyal işleme ve görüntü işleme uygulamalarında kullanılmaktadır. Literatürde mevcut olan medyan, ortalama ve gauss filtrelerinin performansı gözlemlendikten sonra tezde medyan filtre kullanımına karar verilmiştir. Şekil 4.3'te ham bir EMG sinyali ile medyan filtre uygulanmış bir EMG sinyaline ait örnekler gösterilmektedir.

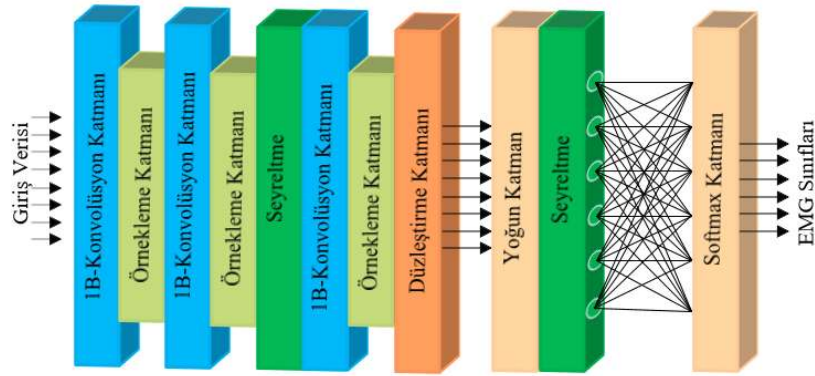


Şekil 4.3. Ham bir EMG sinyali ve medyan filtre uygulanmış bir EMG sinyali örnekleri

4.1.4. Önerilen KSA Modelinin Yapısı

Son on yılda KSA'lar, görüntü işleme problemlerinde öznetelik çıkarımı için en geçerli yaklaşım olmuştur. Standart bir KSA'da model, iki boyutlu verileri işlemektedir [211]. Standart bir KSA sürecindeki adımlar, tek boyutlu verilerin işlenmesinde de izlenebilmektedir. Tek boyutlu KSA ile iki boyutlu KSA arasındaki en belirgin fark, verinin tek boyutlu olarak işlenmesi ve kullanılan filtrelerin veri üzerinde yapmış olduğu işlemidir. Bu çalışmada tasarlanan derin öğrenme modeli, Şekil 4.4'te görüldüğü gibi giriş ve çıkış katmanları dâhil olmak üzere 11 katmandan oluşmaktadır.

Katmanlar arasında 1B-Konvolüsyon (Conv1D), 1B-Maksimum Örnekleme (1D-MaxPooling), seyreltme ve yoğun katman bulunmaktadır. Şekil-4.4'te önerilen modele ait katmanlar gösterilmektedir. Modelin ilk katmanı ile EMG verileri alınmıştır. Sonraki katmanda konvolüsyon işlemi, alınan verilere 3 adım kaydırma, 32 filtre ve 3 eleman kullanılarak uygulanmıştır. Giriş verisinin özellik haritaları konvolüsyon işleminden sonra oluşturulmuştur.



Şekil 4.4. Önerilen 11 katmanlı KSA modelinin blok diyagramı

Özellik haritalarına 1B-Maksimum Örnekleme katmanında maksimum örnekleme uygulanmıştır. Böylece özellik haritaları boyutları, adım kaydırma ve örnekleme boyutu değerleri ile azaltılmıştır. Aşırı öğrenmenin önüne geçmek için rasgele seçilen bazı nöronlar eğitim sürecinde seyreltme yöntemi ile çıkarılmıştır. Önceki katmanlarda verinin özelliklerinin yoğun katmanda kullanılabilir olması için boyut dönüşümleri, düzleştirme (flatten) katmanı yardımı ile gerçekleştirilmiştir. Softmax fonksiyonu, son katmanda kullanılarak sınıflandırma işlemi yapılmıştır. Softmax fonksiyonu yEMG verisini; silindir, hafif tutuş, kanca, avuç içi, küresel ve yanal olmak üzere 6 farklı kategoride sınıflandırmıştır. Önerilen modelin parametreleri Tablo 4.1’de verilmiştir.

Tablo 4.1. Önerilen modelin parametreleri

<i>Katman</i>	<i>Katman tipi</i>	<i>Filtre ve Nöron Sayısı</i>	<i>Diğer Parametreler</i>
1	Conv1D	3x32	Aktivasyon=ReLU, Kaydırma=3
2	MaxPooling1D	—	Örnekleme boyutu=2, Kaydırma=2
3	Conv1D	2x32	Aktivasyon=ReLU, Kaydırma=1
4	MaxPooling1D	—	Örnekleme boyutu=2, Kaydırma=2
5	Dropout	—	Oran=0.5
6	Conv1D	2x32	Aktivasyon=ReLU, Kaydırma=1
7	MaxPooling1D	—	Örnekleme boyutu=2, Kaydırma=2
8	Flatten	—	-
9	Dense	1x256	Aktivasyon=ReLU
10	Dropout	—	Oran=0.5
11	Dense	1x6	Softmax

Genel olarak en iyi performansın sağlanması için farklı verilerin test edilerek parametrelerin ayarlanması gerekmektedir. Ayrıca katman seçiminin ve seçilen katmanların

modelin hangi kısmına yerleştirilmesine yönelik tasarımların yapılması oldukça zaman almaktadır. Hesaplama gücü yüksek donanımlar ve optimizasyon yöntemleri ile bu sorun çözülebilmektedir. Diğer taraftan, en iyi performansın sağlanması için uzman kişiler yardımıyla hiper-parametreler ayarlanabilmektedir. Bu çalışmada, brute-force yöntemi ile hiper-parametreler ve modelin parametreleri seçilmiştir. Her bir parametre, kabaca yapılan tahminlerle en iyi performansı verecek şekilde belirlenmiştir. Optimizasyon için Adam optimizasyon algoritması kullanılmıştır. Yitim fonksiyonu için kategorik-çapraz-düzensizlik yöntemi kullanılmıştır. Modelin hiper-parametrelerinden bazıları yığın boyutu=64, adım sayısı=140 ve öğrenme oranı=0.001 olarak belirlenmiştir. Bu çalışmadaki en önemli yenilik, sınıflandırma için sadece bir kanaldaki verilerin kullanılmasıdır. Bu durum, işlem maliyetinin düşürülmesine önemli katkılar sağlamaktadır. Daha önce yapılan çalışmalarda genellikle birden fazla kanal kullanılmıştır [273,288,300,302,303]. Ayrıca bu uygulamada ön işlemede medyan filtre kullanılmıştır. Bu uygulamada önerilen modele bakıldığında katman sayısının az olmasından dolayı az sayıda hiperparametreden oluşmaktadır.

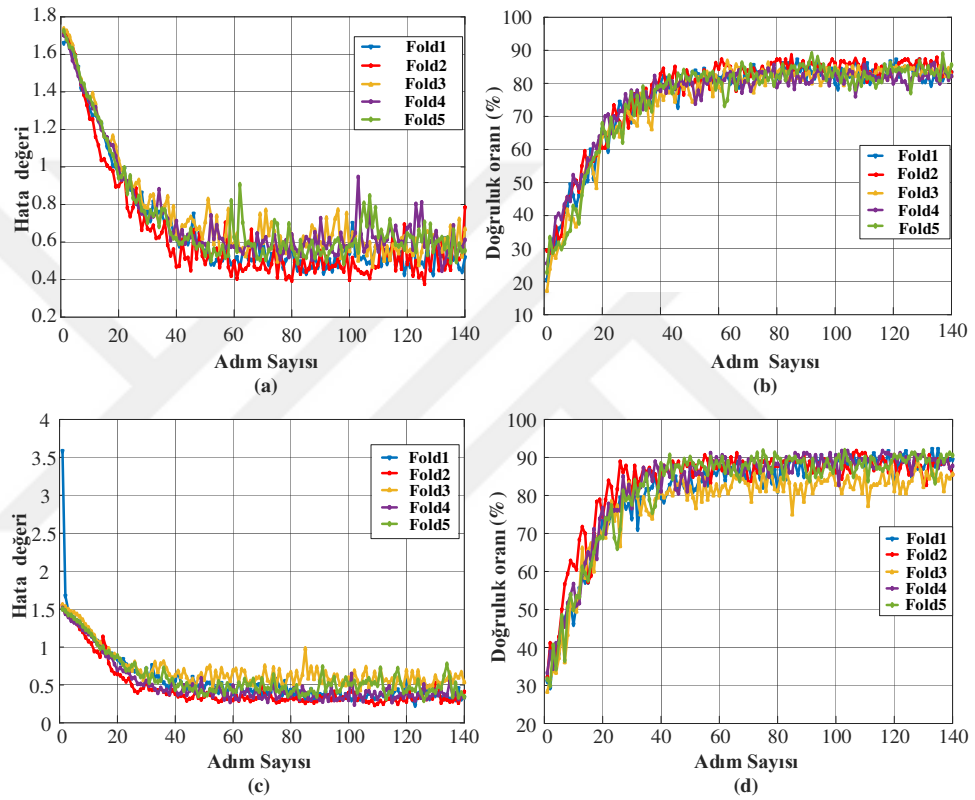
4.1.5. Uygulamalar

Bu çalışmada, 1. veri kümesi ve 2. veri kümesinde bulunan kanal-1 ve kanal-2'den elde edilen sinyaller ayrı ayrı kullanılmıştır. Kanal-1 elektrotu Flexor Capri Ulnaris ve Extensor Capri Radialis, kanal-2 elektrotu ise Longus ve Brevis elektrotu olarak bilinir. 1. veri kümesi ve 2. veri kümesine ek olarak tüm verileri kapsayan kanala göre düzenlenmiş, her iki veri kümesinin karışımı yeni bir veri kümesi oluşturulmuştur ve karışık veri kümesi şeklinde isimlendirilmiştir. Uygulamalarda ek olarak bu veri kümesinin kullanılmasıyla önerilen modelin performansı değerlendirilmiştir. Böylece önerilen modelin üç farklı veri kümesi üzerindeki performansı değerlendirilmiştir. Elde edilen sonuçlar açısından bakıldığında önerilen yöntemin oldukça etkili olduğu görülmektedir. Önileme aşamasında modelin giriş katmanından alınan veriler medyan filtre ile düzeltme işlemine tabi tutulmuştur. Medyan filtre dışında ortalama ve gauss filtrelerin performansına da bakılmıştır. Elde edilen sonuçlardan, medyan filtrenin diğer filtrelerle karşılaştırıldığında kullanılan verilerde daha iyi performans sergilediği görülmüştür. Diğer filtrelerin sonuçları yoğunluk oluşturduğu için verilmemiştir. Ayrıca MinMax ve StandartScaler dönüşümleri, ölçeklendirme için kullanılmıştır. MinMax dönüşümü, tüm verilerin genliklerini genellikle 0 ile 1 aralığına normalize etmektedir. Burada elde edilen dağılım ile verinin dağılımı benzerdir. StandartScaler dönüşümünde ise ortalama değer 0, standart sapma ise 1 değerini alır ve dağılım normale yaklaşır.

Bu çalışmada, verilerin % 80'i eğitim, % 20'si de validasyon için kullanılmıştır. Sonuçlar değerlendirilirken 5-kat (fold) çapraz doğrulama işlemine tabi tutulmuştur. K-kat çapraz doğrulama yöntemi, tüm veriyi alt kümelerle bölüp eğitim ve validasyon süreçlerinden geçirmektedir. Bu durum, modelin performansının rastgele olup olmadığını gösterir. Bu sebeple her bir yitim ve

doğruluk grafiği için her bir kümeye ait sonuçlar ayrı ayrı verilmiştir. Her bir veri kümesi için her kanala ait validasyon seti kullanılarak 5-kat çapraz doğrulama sonuçlarında ortalama (ortanca $\pm$ standart sapma) değerlerine bakılmıştır. Ayrıca performans değerlendirmesi için hata matrisi sonuçları verilmiştir. GİB-tabanlı Tensorflow sunucu uygulaması ve Keras platformu önerilen modelin tasarımı ve uygulamalar için kullanılmıştır.

İlk uygulamada 1. veri kümesine ait kanal-1 ve kanal-2 verileri ayrı ayrı kullanılmıştır. Şekil 4.4'te önerilen modelin, 1. veri kümesinin kanal-1 ve kanal-2'ye ait veriler üzerinde yapılan çalışmalardan elde edilen sonuçlar Şekil 4.5'te gösterilmektedir. Kanal-1 için elde edilen en yüksek

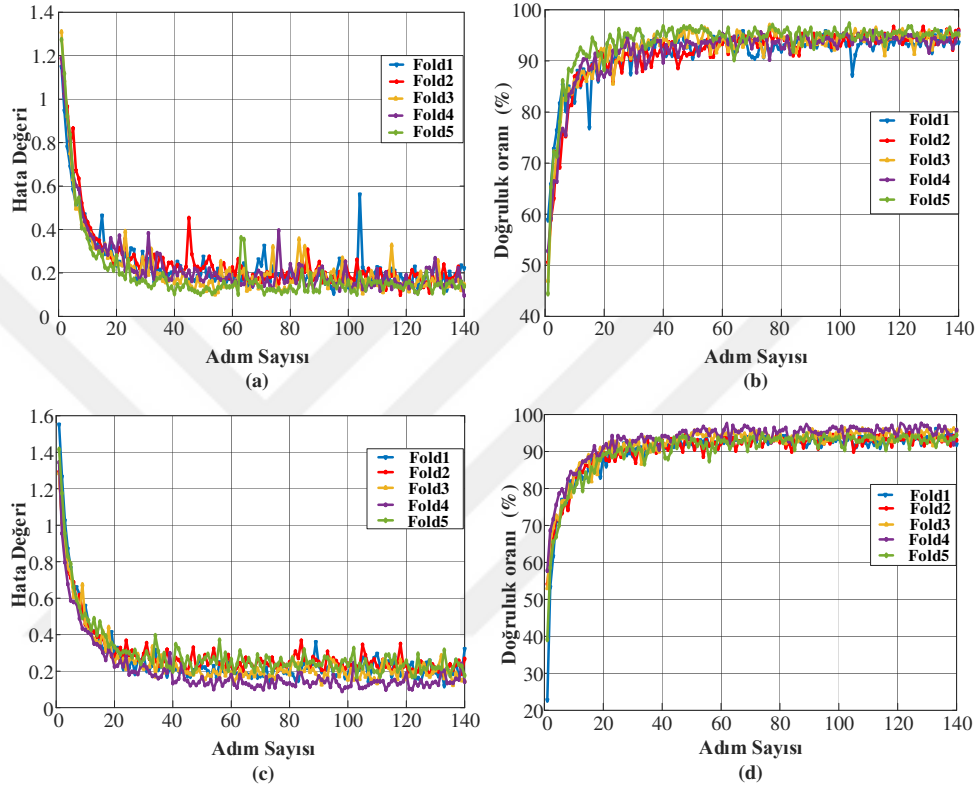


Şekil 4.5. 1. veri kümesindeki kanal-1 (a, b) ve kanal-2'ye (c, d) ait veriler üzerinde yapılan çalışmalardan elde edilen sonuçlar

doğruluk değeri % 89.44 olarak Şekil 4.5 (b)'de görüldüğü gibi Fold5 ile elde edilmiştir. Kanal-2 için elde edilen en yüksek doğruluk değeri ise % 92.22 olarak Şekil 4.5 (d)'de görüldüğü gibi Fold1 ve Fold5 ile elde edilmiştir.

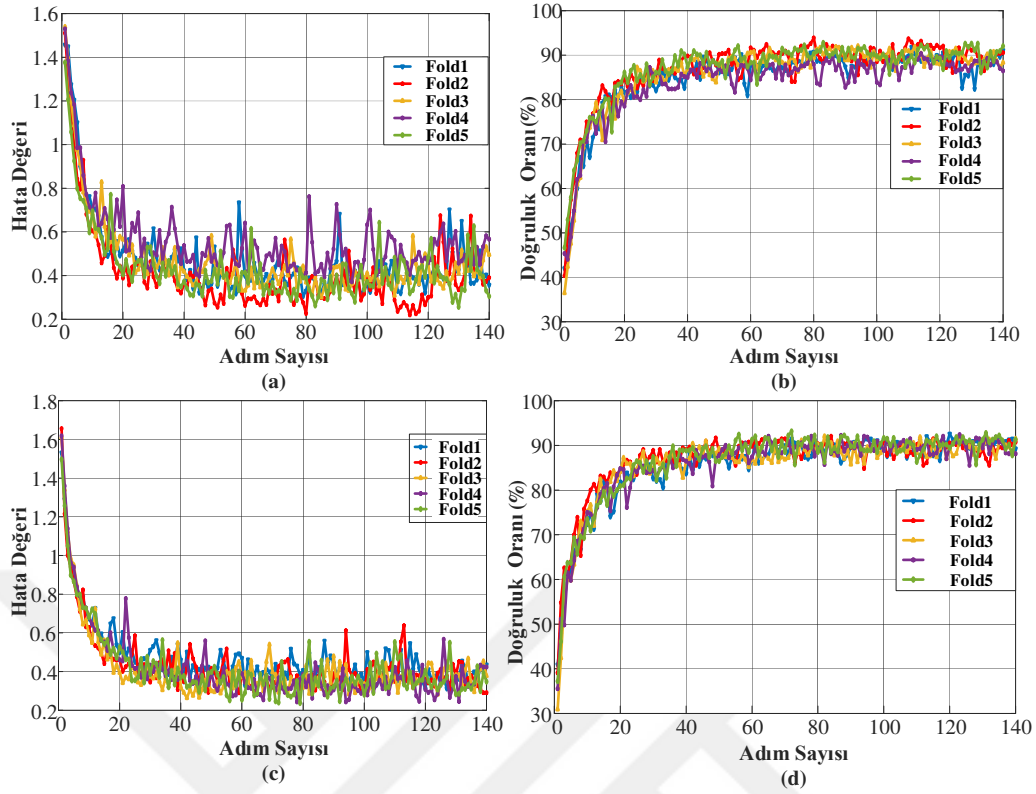
İkinci uygulamada ise 2. veri kümesine ait kanal-1 ve kanal-2 verileri ayrı ayrı kullanılmıştır. Şekil 4.4'te önerilen modelin, 2. veri kümesine ait kanal-1 ve kanal-2 verileri üzerinde yapılan çalışmalardan elde edilen sonuçlar Şekil 4.6'da gösterilmektedir. Elde edilen en yüksek doğruluk değeri % 97.22 olarak Şekil 4.6 (b)'de görüldüğü gibi kanal-1 için Fold5 ile elde edilmiştir. Elde edilen en yüksek doğruluk değeri ise % 97.78 olarak Şekil 4.6 (d)'de görüldüğü gibi kanal-2 için Fold4 ile elde edilmiştir.

Üçüncü uygulamada karışık veri kümesine ait kanal-1 ve kanal-2 verileri kullanılmıştır. Şekil 4.7’de, önerilen modelin karışık veri kümesine ait kanal-1 ve kanal-2 verileri üzerinde yapılan çalışmalardan elde edilen sonuçlar sunulmuştur. Kanal-1 için elde edilen en yüksek doğruluk değeri % 93.89 olarak Şekil 4.7 (b)’de görüldüğü gibi Fold2 ile elde edilmiştir. Kanal-2 için elde edilen en yüksek doğruluk değeri ise % 93.15 olarak Şekil 4.7 (d)’de görüldüğü gibi Fold5 ile elde edilmiştir.



Şekil 4.6. 2. veri kümesine ait kanal-1 (a, b) ve kanal-2 (c, d) verileri üzerinde yapılan çalışmalardan elde edilen sonuçlar

Şekil 4.8’de önerilen modelin 2. veri kümesine ait kanal-1, 1. veri kümesine ait kanal-2 ve karışık veri kümesine ait kanal-2 için elde edilen hata matrisleri verilmiştir. Şekil 4.8 (a)’da görüldüğü gibi 2. veri kümesinin kanal-1 verisi altı kavrama sinyalindeki 360 veri parçasının 342’sini %95.265 doğruluk oranıyla sınıflandırmıştır. Bir başka uygulamada Şekil 4.8 (b)’de görüldüğü gibi 1. veri kümesinin kanal-2 verisi altı kavrama sinyalindeki 180 veri parçasının 160’ını % 89.888 doğrulukta sınıflandırmıştır. Diğer bir uygulamada ise Şekil 4.8 (c)’de görüldüğü gibi karışık veri kümesinin kanal-2 verisi altı kavrama sinyalindeki 510 veri parçasının 487’sini % 90.353 doğrulukta sınıflandırmıştır.



Şekil 4.7. Karışık veri kümesine ait kanal-1 (a, b) ve kanal-2 (c, d) verileri üzerinde yapılan çalışmalardan elde edilen sonuçlar

Modelin performansı; doğruluk, F1-skor, kesinlik, seçicilik ve duyarlılık değerleri ile 5-kat çapraz doğrulama yöntemiyle hesaplanmıştır. Performans ölçütleri; TP (True Positive – Doğru Pozitif), TN (True Negative – Doğru Negatif), FP (False Positive – Yanlış Pozitif) ve FN (False Negative – Yanlış Negatif) örnekleri kullanılarak hesaplanmıştır [304–307].

$$\text{Doğruluk} = (TP + TN) / (TP + TN + FP + FN)$$

$$F - \text{skor} = 2 \times (\text{Kesinlik} \times \text{Duyarlılık}) / (\text{Kesinlik} + \text{Duyarlılık})$$

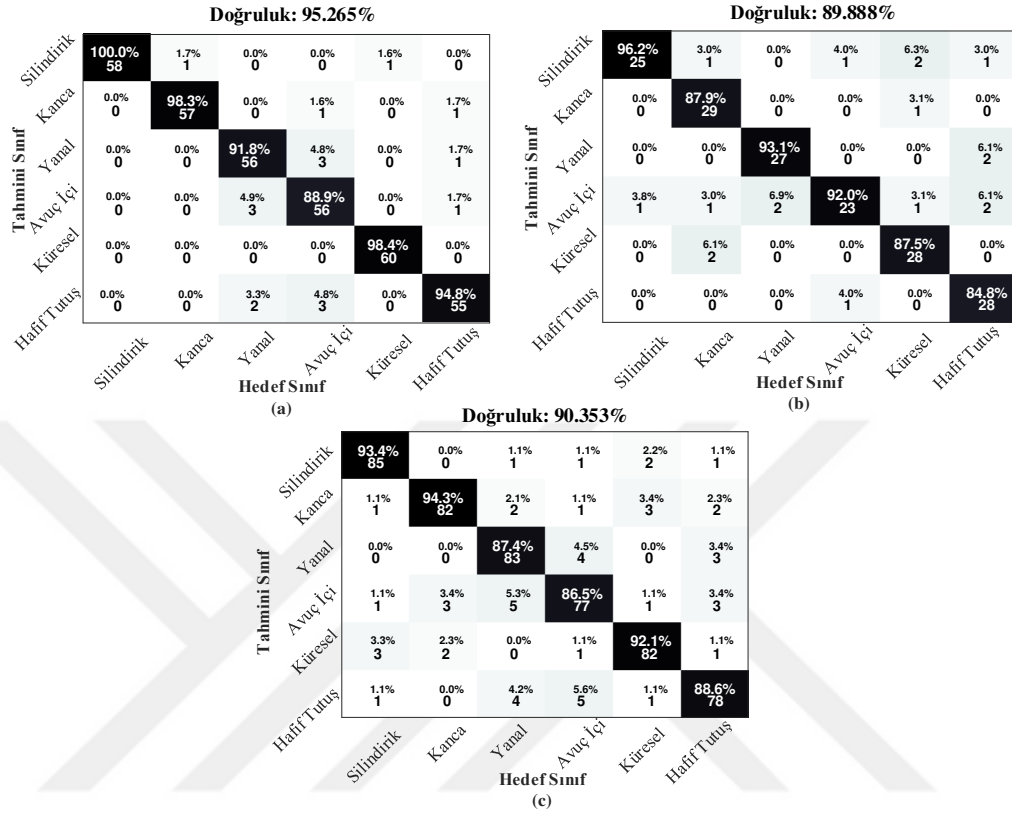
$$\text{Kesinlik} = TP / (TP + FP)$$

$$\text{Duyarlılık} = TP / (TP + FN)$$

$$\text{Seçicilik} = TN / (FP + FN)$$

Önerilen modelin performans parametrelerinin ortalama ve standart sapma değerleri, veri kümelerinin her bir kanalı için Tablo 4.2’den Tablo 4.7’ye kadar verilmiştir. Tablo 4.2, 1. veri kümesinin kanal-1 doğrulama setini kullanan 5 kat çapraz doğrulama stratejisiyle önerilen modelin ortalama değerlerini göstermektedir. En yüksek doğruluk değerini %95.55 ile yanal kavrama tipi, en düşük değeri ise %93.33 ile avuç içi elde etmiştir. En yüksek kesinlik değerini %90.91 ile silindirik kavrama tipi, en düşük değeri ise küresel kavrama tipi %76.59 ile elde etmiştir. En düşük duyarlılık değeri silindirik kavrama tipi için %75.10, en yüksek değer ise hafif tutuş kavrama tipi için %94.17’dir. En yüksek seçicilik değerini %98.23 ile silindirik kavrama tipi, en düşük değeri

ise %94.29 ile küresel kavrama tipi elde etmiştir. En düşük F1-skör değeri avuç içi kavrama tipi için %78.92, en yüksek değer ise hafif tutuş kavrama tipi için %87.47'dir.



Şekil 4.8. (a) 2. veri kümesine ait kanal-1, (b) 1. veri kümesine ait kanal-2 ve (c) karışık veri kümesine ait kanal-2 için elde edilen hata matrisleri

Tablo 4.2. Doğrulama seti olarak 1. veri kümesinin kanal-1'i kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans

Sınıflar	Doğruluk(%)	Kesinlik(%)	Duyarlılık(%)	Seçicilik(%)	F1-skör
Silindirik	94.33 $\pm$ 2.30	90.91 $\pm$ 6.82	75.10 $\pm$ 10.55	98.23 $\pm$ 1.55	81.80 $\pm$ 6.24
Yanal	95.55 $\pm$ 0.55	85.81 $\pm$ 7.44	87.26 $\pm$ 3.37	97.36 $\pm$ 0.92	86.28 $\pm$ 2.69
Kanca	94.55 $\pm$ 1.49	86.06 $\pm$ 8.22	81.26 $\pm$ 4.02	97.19 $\pm$ 1.74	83.38 $\pm$ 3.93
Avuç İçi	93.33 $\pm$ 2.25	85.13 $\pm$ 11.66	75.11 $\pm$ 10.96	97.23 $\pm$ 2.43	78.92 $\pm$ 6.83
Küresel	93.44 $\pm$ 1.54	76.59 $\pm$ 9.18	90.26 $\pm$ 9.12	94.29 $\pm$ 2.94	82.09 $\pm$ 2.88
Hafif Tutuş	95.44 $\pm$ 1.38	82.11 $\pm$ 7.06	94.17 $\pm$ 4.48	95.72 $\pm$ 1.91	87.47 $\pm$ 3.20

Tablo 4.3, 1. veri kümesinin kanal-2 doğrulama setini kullanan 5 kat çapraz doğrulama stratejisiyle önerilen modelin ortalama değerlerini göstermektedir. Yanal kavrama tipi %97.22 ile en yüksek doğruluk değerine sahipken, en düşük değeri %95.00 ile avuç içi kavrama tipi elde etmiştir. Ulaşılan en yüksek kesinlik değeri kanca kavrama tipi için %92.97, elde edilen en düşük kesinlik değeri ise uç kavrama tipi için %83,77'dir. Elde edilen en düşük duyarlılık değeri avuç içi kavrama tipi için %77.79 ve elde edilen en yüksek değer yanal kavrama tipi için %96.69'dur. En

yüksek seçicilik değerini %98.66 ile avuç içi kavrama tipi, en düşük değeri ise %96.40 ile hafif tutuş elde etmiştir. En düşük F1-skor değeri %83.72 ile avuç içi kavrama tipi için, en yüksek değer ise kanca kavrama tipi için %91.60'dır.

Tablo 4.3. Doğrulama seti olarak 1. veri kümesinin kanal-2'si kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans

Sınıflar	Doğruluk(%)	Kesinlik(%)	Duyarlılık(%)	Seçicilik(%)	F1-skor
Silindirik	95.77 $\pm$ 1.69	91.54 $\pm$ 7.37	81.39 $\pm$ 13.45	98.54 $\pm$ 1.26	85.62 $\pm$ 7.95
Yanal	97.00 $\pm$ 1.90	87.34 $\pm$ 12.60	96.69 $\pm$ 5.31	97.25 $\pm$ 2.58	91.14 $\pm$ 6.46
Kanca	97.22 $\pm$ 2.25	92.97 $\pm$ 10.15	90.59 $\pm$ 4.47	98.54 $\pm$ 2.14	91.60 $\pm$ 6.72
Avuç İçi	95.00 $\pm$ 2.60	92.35 $\pm$ 2.80	77.79 $\pm$ 13.72	98.66 $\pm$ 0.66	83.72 $\pm$ 7.89
Küresel	96.44 $\pm$ 2.47	86.80 $\pm$ 9.82	93.60 $\pm$ 5.89	97.06 $\pm$ 2.24	89.85 $\pm$ 6.56
Hafif Tutuş	95.88 $\pm$ 0.49	83.77 $\pm$ 3.01	93.69 $\pm$ 4.66	96.40 $\pm$ 0.53	88.33 $\pm$ 1.07

Tablo 4.4, 2. veri kümesinin kanal-1 doğrulama setini kullanan 5 kat çapraz doğrulama stratejisiyle önerilen modelin ortalama değerlerini göstermektedir. Küresel kavrama tipi %99.94 ile en yüksek doğruluk değerine sahiptir ve avuç içi %96.83 ile en düşük doğruluk değerine sahiptir. Küresel kavrama tipi %99.66 ile en yüksek kesinlik değerine, avuç içi ise %90.38 ile en düşük kesinlik değerine ulaşmıştır. Ortaya çıkan en düşük duyarlılık değeri avuç içi kavrama tipi için %90.28 ve küresel kavrama tipi için en yüksek değer %100'dür. En yüksek seçicilik değerini %99.93 ile küresel kavrama tipi, en düşük değeri ise %98.06 ile avuç içi elde etmiştir. En yüksek F1-skor değeri küresel kavrama tipi için %99.82, en düşük değer ise avuç içi kavrama tipi için %90.17'dir.

Tablo 4.4. Doğrulama seti olarak 2. veri kümesinin kanal-1'i kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans

Sınıflar	Doğruluk(%)	Kesinlik(%)	Duyarlılık(%)	Seçicilik(%)	F1-skor
Silindirik	99.33 $\pm$ 0.63	97.19 $\pm$ 2.65	98.52 $\pm$ 1.59	99.47 $\pm$ 0.50	97.85 $\pm$ 2.09
Yanal	99.27 $\pm$ 0.24	97.84 $\pm$ 1.91	97.92 $\pm$ 1.37	99.52 $\pm$ 0.53	97.86 $\pm$ 0.49
Kanca	97.61 $\pm$ 0.82	94.21 $\pm$ 4.47	91.36 $\pm$ 3.60	98.87 $\pm$ 0.88	92.67 $\pm$ 2.61
Avuç İçi	96.83 $\pm$ 0.75	90.38 $\pm$ 1.69	90.28 $\pm$ 7.37	98.06 $\pm$ 0.51	90.17 $\pm$ 3.50
Küresel	99.94 $\pm$ 0.12	99.66 $\pm$ 0.75	100.00 $\pm$ 0.00	99.93 $\pm$ 0.14	99.82 $\pm$ 0.38
Hafif Tutuş	97.33 $\pm$ 0.63	91.99 $\pm$ 5.08	92.39 $\pm$ 3.77	98.33 $\pm$ 1.08	92.04 $\pm$ 1.74

Tablo 4.5, 2. veri kümesinin kanal-2 doğrulama setini kullanan 5 kat çapraz doğrulama stratejisiyle önerilen modelin ortalama değerlerini göstermektedir. Silindirik kavrama tipi %99.55 ile en yüksek doğruluk değerine sahipken, en düşük değere %96.44 ile avuç içi kavrama tipi sahiptir. En yüksek hassasiyet değeri %99.10 ile silindirik kavrama tipi için, en düşük hassasiyet

değeri ise %89.66 ile kanca tipi için olmuştur. Elde edilen en düşük duyarlılık değeri avuç içi kavrama tipi için %84.33, elde edilen en yüksek değer ise küresel kavrama tipi için %98.68'dir. En yüksek seçicilik değerini %99.79 ile silindirik kavrama tipi, en düşük değeri ise %97.72 ile kanca elde etmiştir. En düşük F1-skor değeri avuç içi kavrama tipi için %88.67, en yüksek değer silindirik kavrama tipi için %98.60'tır.

Tablo 4.5. Doğrulama seti olarak 2. veri kümesinin kanal-2'si kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans

Sınıflar	Doğruluk(%)	Kesinlik(%)	Duyarlılık(%)	Seçicilik(%)	F1-skor
Silindirik	99.55 $\pm$ 0.50	99.10 $\pm$ 1.30	98.16 $\pm$ 3.33	99.79 $\pm$ 0.30	98.60 $\pm$ 1.70
Yanal	98.66 $\pm$ 1.13	95.05 $\pm$ 5.53	97.95 $\pm$ 2.10	98.77 $\pm$ 1.68	96.37 $\pm$ 2.33
Kanca	97.33 $\pm$ 0.91	89.66 $\pm$ 4.34	95.24 $\pm$ 2.91	97.72 $\pm$ 1.21	92.29 $\pm$ 2.28
Avuç İçi	96.44 $\pm$ 0.88	94.33 $\pm$ 4.62	84.33 $\pm$ 8.38	98.85 $\pm$ 1.03	88.67 $\pm$ 3.05
Küresel	99.44 $\pm$ 0.39	98.16 $\pm$ 2.44	98.68 $\pm$ 1.29	99.59 $\pm$ 0.55	98.40 $\pm$ 1.01
Hafif Tutuş	96.88 $\pm$ 1.04	90.96 $\pm$ 5.17	90.27 $\pm$ 2.58	98.20 $\pm$ 1.04	90.56 $\pm$ 3.33

Tablo 4.6, karışık veri kümesinin kanal-1'i doğrulama seti olarak kullanılarak 5 kat çapraz doğrulama stratejisiyle önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performansı sunulmuştur. En yüksek doğruluk değerine yanıl kavrama tipi %97.67 olarak ve en düşük değere ise %95.37 ile kanca kavrama tipi sahiptir. En düşük kesinlik değeri kanca kavrama tipi için %84.91, en yüksek kesinlik değeri yanıl tip için %95.44'tür. Elde edilen en düşük duyarlılık değeri avuç içi kavrama tipi için %85.47 ve en yüksek değer ise küresel kavrama tipi için %95.69'dur. En düşük seçicilik değerini %96.74 ile kanca kavrama tipi, en yüksek değeri ise %99.16 ile yanıl kavrama tipi elde etmiştir. En yüksek F1-skor değeri yanıl kavrama tipi için %92.71 ve en düşük değer ise avuç içi kavrama tipi için %86.24'tür.

Tablo 4.6. Doğrulama seti olarak karışık veri kümesinin kanal-1'i kullanılarak 5-kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans

Sınıflar	Doğruluk(%)	Kesinlik(%)	Duyarlılık(%)	Seçicilik(%)	F1-skor
Silindirik	97.07 $\pm$ 0.73	93.06 $\pm$ 3.29	89.23 $\pm$ 4.02	98.67 $\pm$ 0.62	91.03 $\pm$ 2.24
Yanal	97.67 $\pm$ 0.55	95.44 $\pm$ 3.23	90.21 $\pm$ 3.07	99.16 $\pm$ 0.56	92.71 $\pm$ 2.05
Kanca	95.37 $\pm$ 1.79	84.91 $\pm$ 6.34	88.53 $\pm$ 4.70	96.74 $\pm$ 1.59	89.45 $\pm$ 4.83
Avuç İçi	95.48 $\pm$ 1.67	87.19 $\pm$ 5.25	85.47 $\pm$ 7.10	97.46 $\pm$ 1.07	86.24 $\pm$ 5.47
Küresel	97.48 $\pm$ 0.77	90.06 $\pm$ 2.44	95.69 $\pm$ 4.45	97.96 $\pm$ 0.33	92.70 $\pm$ 1.40
Hafif Tutuş	96.33 $\pm$ 1.18	88.40 $\pm$ 6.57	89.75 $\pm$ 2.07	97.65 $\pm$ 1.27	88.99 $\pm$ 3.87

Tablo 4.7, doğrulama seti olarak karışık veri kümesinin kanal-2'sini kullanan 5 kat çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama performansı göstermektedir. En yüksek doğruluk değerini %97.96 ile silindirik kavrama tipi elde ederken, en düşük değeri %95.26 ile avuç içi kavrama tipi elde etmiştir. Avuç içi kavrama tipi için en düşük kesinlik değeri %86.11,

en yüksek kesinlik değeri silindirik tip için %93.73'tür. Elde edilen en düşük duyarlılık değeri avuç içi kavrama tipi için %85.66, elde edilen en yüksek değer ise silindirik kavrama tipi için %94.50'dir. En yüksek seçicilik değerini %98.66 ile silindirik kavrama tipi, en düşük değeri ise %97.14 ile avuç içi elde etmiştir. En düşük F1-skor değeri avuç içi kavrama tipi için %85.72, en yüksek değer ise silindirik kavrama tipi için %94.06'dır.

Tablo 4.7. Doğrulama seti olarak karışık veri kümesinin kanal-2'si kullanılarak 5-fold çapraz doğrulama stratejisi ile önerilen model için elde edilen ortalama (ortalama $\pm$ standart sapma) performans

Sınıflar	Doğruluk(%)	Kesinlik(%)	Duyarlılık(%)	Seçicilik(%)	F1-skor
Silindirik	97.96 $\pm$ 1.07	93.73 $\pm$ 4.55	94.50 $\pm$ 2.71	98.66 $\pm$ 1.04	94.06 $\pm$ 2.87
Yanal	97.22 $\pm$ 0.57	92.61 $\pm$ 5.88	91.00 $\pm$ 3.14	98.50 $\pm$ 1.24	91.62 $\pm$ 1.41
Kanca	96.41 $\pm$ 0.59	87.32 $\pm$ 3.28	91.79 $\pm$ 2.99	97.34 $\pm$ 0.69	89.45 $\pm$ 1.99
Avuç İçi	95.26 $\pm$ 1.31	86.11 $\pm$ 4.60	85.66 $\pm$ 6.92	97.14 $\pm$ 1.17	85.72 $\pm$ 3.91
Küresel	97.37 $\pm$ 0.92	92.53 $\pm$ 3.95	91.14 $\pm$ 6.33	98.64 $\pm$ 0.62	91.72 $\pm$ 3.85
Hafif Tutuş	96.15 $\pm$ 1.90	91.02 $\pm$ 10.70	86.78 $\pm$ 4.79	97.94 $\pm$ 2.53	88.44 $\pm$ 4.81

5 kat çapraz doğrulama sonuçlarına bakıldığında Tablo 4.4'te görüldüğü gibi en yüksek doğruluk değeri % 99.94 olarak 2. veri kümesi kanal-2 kullanılarak küresel kavrama tipi için elde edilmiştir. En düşük doğruluk oranı is Tablo 4.2'de görüldüğü gibi % 93.33 olarak avuç içi kavrama tipi için 1. veri kümesi kanal-1'den alınan sinyaller ile elde edilmiştir. En yüksek kesinlik değeri % 99.66 ile Tablo 4.4'te verildiği gibi 2. veri kümesi kanal-1'e ait sinyallerden küresel kavrama tipi için elde edilmiştir. En düşük kesinlik değeri, % 82.11 olarak Tablo 4.2'de görüldüğü gibi 1. veri kümesinin kanal-1'i kullanılarak hafif tutuş kavrama tipi için elde edilmiştir. En yüksek duyarlılık değeri, % 100 ile 2. veri kümesi kanal-1 sinyalleri kullanılarak Tablo 4.4'te verildiği gibi küresel kavrama tipi için elde edilmiştir. En düşük duyarlılık değeri ise silindir kavrama tipine ait 1. veri kümesi kanal-1 sinyalleri ile Tablo 4.2'de gösterildiği gibi % 75.10 olarak bulunmuştur. En yüksek seçicilik değeri, 2. veri kümesi kanal-1 küresel kavrama sinyalleri için Tablo 4.4'te gösterildiği gibi % 99.93 olarak elde edilmiştir. En düşük seçicilik değeri, % 94.29 olarak Tablo 4.2'de görüldüğü gibi 1. veri kümesinin kanal-1'i kullanılarak küresel kavrama tipi için elde edilmiştir. En yüksek F1-skor değeri % 99.82 ile Tablo 4.4'te verildiği gibi 2. veri kümesi kanal-1'e ait sinyallerden küresel kavrama tipi için elde edilmiştir. En düşük F1-skor değeri ise avuç içi kavrama tipine ait 1. veri kümesi kanal-1 sinyalleri ile Tablo 4.2'de gösterildiği gibi % 78.92 olarak bulunmuştur.

4.1.6. Tartışma

Kavramaya yönelik EMG sinyallerinin sınıflandırılmasında ilgili literatür 4.1.1 alt bölümünde bahsedildiği gibi birçok makine öğrenmesi algoritması kullanılmıştır. Bu çalışmalarda genellikle geleneksel öznitelik çıkarma ve sınıflandırma yöntemleri kullanılmıştır. Alt bölüm

4.1.1’de belirtildiği gibi farklı çalışmalarda farklı sayıda kanal ve kavrama tipleri kullanılmıştır. Tablo 4.8’de aynı veri kümesi ile farklı yEMG sinyallerini kullanarak kavrama tiplerini sınıflandıran öne çıkan farklı çalışmaların performansları ve özellikleri karşılaştırılmıştır.

Tablo 4.8. Farklı çalışmaların performansları ve özellikleri karşılaştırılması

Çalışma	Veri Kümesi ve Kanal Sayısı	Öznitelik Çıkartma Yöntemi	Sınıflandırma Yöntemi	Accuracy (%)
Referans-1[288]	Aynı & 2 Kanal	Empirical Mode Decomposition	Basit bir doğrusal sınıflandırıcısı	86.64
Referans-2[273]	Aynı & 2 Kanal	Tekil Değer Ayırıştırma ve TBA	K-En yakın komşuluk sınıflandırıcısı	86.71
Referans-3[302]	Aynı & 2 Kanal	Derin Orman (Deep Forest)		71.3
Referans-4[300]	Aynı & 2 kanal	TQWT + Kraskov Entropy	K-En yakın komşuluk sınıflandırıcısı	98.55
Referans-5[303]	Aynı & 2 Kanal	Sıfır Geçişler	DVM	85.66
Önerilen Yöntem	Aynı & 1 Kanal	1B-KSA	Softmax Sınıflandırıcı	94.94

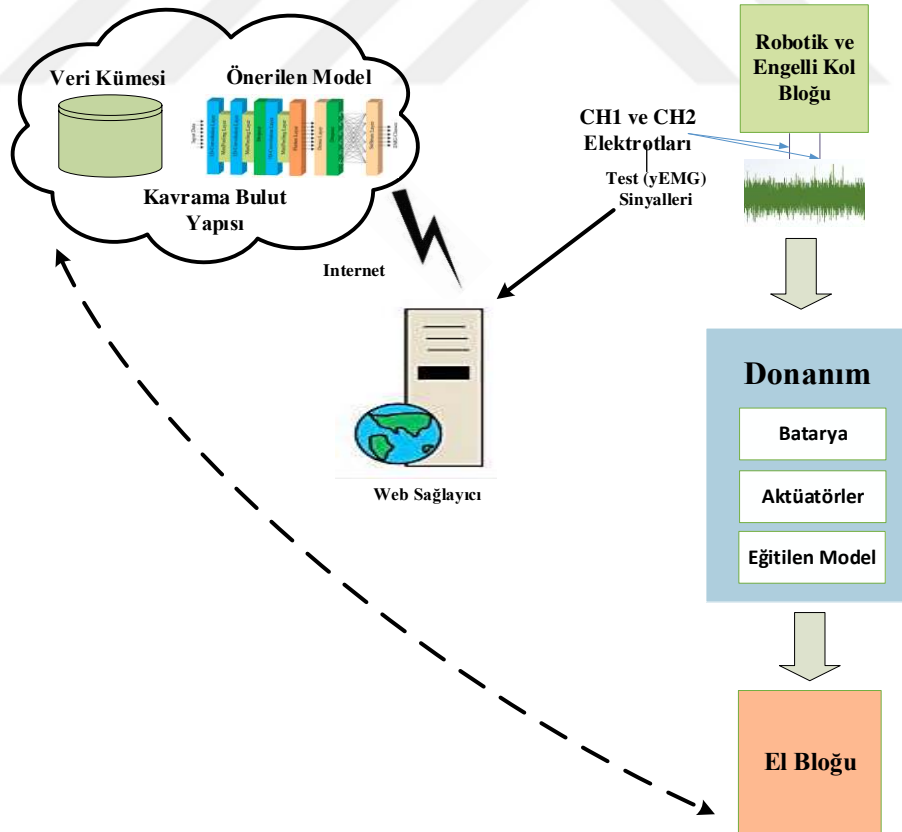
Sapsanis ve arkadaşları, ham EMG sinyallerinden çıkartılan özniteliklerin ayırıştırılmasını geliştirmek için deneysel modlu ayırıştırma tekniğini kullanmışlardır [288]. Önermiş oldukları yöntemi 1. veri kümesi üzerinde test ederek basit bir sınıflandırıcı ile % 86.64 doğrulukta bir sonuca ulaşmışlardır. Iqbal ve arkadaşları, temel bileşenler analizini ve tekil değer ayırıştırma yöntemlerini öznitelik çıkartmada ve KYK sınıflandırıcısı ise sınıflandırmada kullanmışlardır [273]. Bu yöntemle % 86.71 oranında bir doğruluk elde etmişlerdir. Zhou ve Feng tarafından yapılan çalışmada, gcForest yaklaşımını kullanarak % 71.3 oranında sınıflandırma başarımı elde etmişlerdir [302]. Toledo-Perez ve arkadaşları, sıfır geçişli bir yöntemi öznitelik çıkarırken ve DVM’yi sınıflandırma yaparken kullanmışlardır [303]. Önermiş oldukları yöntem ile % 85.66 doğruluk ile sınıflandırma yapmışlardır. Nishad ve arkadaşları tarafından yapılan çalışmada ise ayarlanabilir-Q dalgacık dönüşümü ve Kraskov entropi yöntemi ile öznitelik çıkarma, KYK sınıflandırıcısı ile sınıflandırma yapmışlardır [300]. Önermiş oldukları yöntem % 98.55 doğrulukta kavrama sinyallerini sınıflandırmıştır.

Yukarıda sonuçları verilen modern yaklaşımlarda, Zhou ve Feng’in tarafından yapılan çalışma hariç genellikle geleneksel makine öğrenmesi yöntemleri kullanılmıştır [302]. Önerilen

yöntemlerde genellikle öznitelik çıkarmaya yönelik iyileştirmeler yapılmaya çalışılmıştır. Ancak bu yöntemlerin tasarımı çok fazla zaman alabilmektedir. Bu tez çalışmasında ise, 1B-KSA öznitelik çıkarmada ve softmax fonksiyonu sınıflandırmada kullanılmıştır.

Nishad ve arkadaşlarının yaptığı çalışma, az bir farkla daha yüksek bir doğruluk ile sınıflandırma performansı gösterse de, öznitelik çıkarma açısından önermiş oldukları geleneksel yöntemin geliştirilmesi hem uğraştırıcı hem de oldukça zaman almaktadır [300]. Ayrıca yaptıkları çalışmada genellikle veri kümesinde yer alan bireylere göre sinyallerin analizini yapmaya çalışmışlardır. Bu uygulamada önerilen yöntem ise yapay zekâda kullanılan en iyi yaklaşımlardan biri olması ve uçtan uca model geliştirmede süreci hızlandırması sebebiyle ön plana çıkmaktadır. Diğer taraftan veri kümesinde yer alan tüm sinyaller karıştırılarak bireylerin esas alınmasından ziyade sinyaller temel alınarak bu uygulama çalışması yapılmıştır. Bu durum, klinik çalışmalarında daha etkili bir yöntem olabilecek olması sebebiyle önemlidir. Bunlara ilaveten, yapmış olduğumuz bu uygulamada tek bir kanal verisi kullanılırken diğer yöntemlerde iki kanal verisi kullanılmıştır.

Gerçek dünyada önerilen yöntemin kullanımına yönelik bir gösterim Şekil 4.9'da görülmektedir. Bu uygulamada önermiş olduğumuz yöntem, nesnelerin interneti ile kullanılarak robotlarda ve engellilerde kavrama becerilerinin geliştirilmesini sağlayabilecek potansiyele sahiptir. Şekil 4.9'e bakıldığında bir bulut sisteminde eğitilen bir model bulunmaktadır.



Şekil 4.9. Bir bulut sisteminde eğitilen derin öğrenme modeli

Aynı model el üzerinde yer alan donanımda da bulunmaktadır. Robot ya da engelli birey bir kavrama hareketini gerçekleştirmek istediği zaman, test sinyalleri hem sunucuya hem de donanıma aynı anda gönderilmektedir. Donanımda eğitilen model, kavrama sinyalinin hangi sınıfa ait olduğunu tespit edebilmektedir. Bu da robot ya da engelli bireyin el hareketini gerçekleştirebilmesini sağlamaktadır. Ayrıca test sinyalleri buluttaki sunucuya gönderilmektedir. Alınan yeni yEMG verisinin sınıfı, bulut içerisinde eğitilen model yardımıyla belirlendikten sonra bulutta saklanmaktadır. Bulutta kaydedilen yEMG verisi modeli eğitmek için kullanılabilir. Böylece modelin performansı sürekli olarak geliştirilebilmektedir.

4.1.7. Sonuçlar

Bu uygulamanın amacı, yEMG sinyalleri kullanılarak 6 farklı kavrama tipinin sınıflandırılması için yeni bir derin 1B-KSA modelini geliştirmektir. Medyan filtre, yEMG sinyalleri düzeltmek için kullanılmıştır. 1B-KSA genel olarak ortalama % 94.94 doğrulukla sınıflandırma başarımı göstermiştir. Geliştirilen bu sistem, 4.1.6 alt bölümündeki referans çalışmalarla karşılaştırıldığında klinik çalışanlarının hastalık tahmininin daha doğru yapılmasını sağlayacağı ön görülmektedir. Geliştirilen model gelecekte daha fazla veri ile test edilebilecek potansiyele sahip olup kas sistemi ile ilgili rahatsızlıkların çözümünde kullanılabilir.

4.2. DKQÖ Algoritması ile Pong Oyunu

Bu uygulamada, nesne kavrama uygulamasına hazırlık olarak oyun oynayabilen DKQÖ tabanlı bir yapay zekâ ajanının eğitilmesi süreci öncesinde ele alınmıştır.

Yapay zekâ, insan müdahalesi olmadan kendi kararlarını vermek/öğrenmek/düşünmek için insandaki düşünce sürecinin özelliklerini taklit etmeye çalışan bir sistemdir. Video oyunlarında yapay zekâ, çok temel sistemler olmasına rağmen ilk olarak Pong ve Space Invaders gibi klasik Atari oyunlarında görülmüştür [249,308,309]. Yakın zamanda yapılan yapay zekâ araştırmalarında, derin öğrenme algoritmalarıyla donatılmış bilgisayarların dünyanın en iyi DotA 2 oyuncularından bazılarını yenmiştir. Yapay zekâ ve derin öğrenmede büyük atılımlar beklenmekte ve bilim kurgu klasiklerinde tasvir edilen robotlar yakın gelecekte kurgusal olmaktan çıkmaya doğru yol almaktadır. Araştırmacılara göre, makine öğrenimi ile ilgili gelecekteki araştırmalar, sinir ağlarının otonom olarak eğitilmesiyle sonuçlanacak ve bu da daha fazla sayıda "kendi kendine düşünen" makinelerin üretilmesine yol açacaktır. Bu alt bölümde, oyun oynayabilen DKQÖ tabanlı bir yapay zekâ ajanının/modelinin eğitilmesi süreci ele alınmaktadır.

4.2.1. İlgili Literatür

Günümüzde, yapay zekâ iyi bir video oyunu için temel bir unsur olarak görülmektedir. Xu'ya göre [310] ilk video oyunları, aşırı basit olmaları ve iki veya daha fazla kişi arasında rekabet edecek şekilde tasarlanmaları nedeniyle yapay zekâ olmadan oluşturulmuştur. Ancak yapay zekâ geliştirme tasarımcıları, bilgisayar tarafından bir düşman oluşturarak kişinin tek başına oynaması sağlanmıştır. Bunu yapan ve bugün hala oynanan en dikkate değer oyun Pong oyunu olmuştur [249]. Video oyunlarında yapay zekâ, işlevsellik ve insan benzeri yetenekler açısından uzun bir yol kat etmiştir ancak henüz düşünüldüğü kadar gelişmiş değildir. [311,312]'deki yazarlar, son yılların video oyunu geliştiricilerinin öncelikle oyunları grafiksel olarak mükemmel ve gerçekçi hale getirmeye, rakiplere ayak uydurmaya ve günümüzün talebini karşılamaya odaklandıklarını belirtmektedir. Bu nedenle geliştiriciler, kalıcı bir oyuncu kitlesi oluşturmak için grafik motoruna daha fazla kaynak ayırırken yapay zekâyı daha az kaynak ayırmaktadır.

[310]'nin yazarları ayrıca video oyunu yapay zekâ gelişiminin eksikliğinin üç ana nedeninin olduğunu ifade etmektedirler. İlk olarak, [310]'de yürütülen çalışmalarda MİB gücü ve kaynaklarının eksikliğinden bahsedilmektedir. Grafikler gibi diğer oyun geliştirme alanları daha gelişmiş hale gelmeye devam edeceğinden, daha yüksek düzeyde MİB gücüne ihtiyaç duyulmaktadır. Ancak, daha güçlü GİB'lerin geliştirilmesinden bu yana, bu ağırlık neredeyse tamamen MİB'den ziyade tamamen GİB'ye yönelmeye başlamıştır [313]. İkincisi, güvenilir bir sistem oluşturmak için yapay zekâ programlamasının ve yeteneklerinin mükemmel bir şekilde anlaşılması gerektiğinden, gelişmiş yapay zekâ tekniklerinin anlaşılmasından da kaynaklanabilmektedir. Üçüncüsü ve daha olası olmayan, sinir ağlarını veya diğer deterministik olmayan yöntemleri kullanmanın, oyunu tamamen bozma açısından gerçek bilinmeyen uzun vadeli etkisidir.

[314]'te, video oyunlarının karmaşıklıkları, belirlenemez olmaları ve sınırlı girdileri nedeniyle farklı yapay zekâ algoritmalarını test etmek için yaygın olarak kullanıldığı belirtilmektedir. Togelius'a göre, yapay zekâyı binlerce örnekle oluşturulmuş oyun ortamlarında geliştirmek ve test etmek, robotlar inşa edip onlara binlerce test yaptırmaya göre maliyet açısından çok daha uygundur [308]. Kahn, sürücüsüz araçlar gibi gerçek dünya kullanımları için yazılım tasarlarken, donanım zarar verme ve insanları yaralama riskinden ziyade yapay zekâyı bir bilgisayar oyunu aracılığıyla eğitmenin çok daha kolay olduğunu belirtmektedir [309]. Yakın tarihli bir Forbes makalesinde, Google'ın DeepMind ve Unity Technologies gibi önde gelen şirketlerle yapılan ortaklıkların, DPÖ gibi yapay zekâ yöntemlerini bir sonraki düzeye taşıyabileceğini belirtilmektedir [315]. Unity ortamında, sürücüsüz araçlar ve 3B navigasyon gibi sorunlarının üstesinden gelebilecek yapay zekâ oluşturulmuştur. Video oyunları oynayabilen yapay zekâ, hataları bulmak ve genel olarak oyun kalitesini artırmak için test aşamalarında kullandığı bir makine ile geliştiricilere yardımcı olabilmektedir.

Yannakakis ve Togelius'a göre [316] yapay zekânın video oyunları oynamasındaki ilk büyük kilometre taşlarından biri, 1997'de Minimax algoritmasının, değerlendirme fonksiyonları ve bir süper bilgisayar yardımıyla satranç dünya şampiyonu Garry Kasparov'u mağlup ettiği IBM'in Deep Blue oyunu olmuştur. Sonrasında DeepMind'in antik Çin oyunu Go'ya adanmış yapay zekâ programı AlphaGo, 2016'da Go'nun en iyi oyuncularından bazılarını yenmeyi başarmıştır. Daha fazla olası pozisyonu olan ve strateji gerektiren Go oyununun satrançtan biraz daha zor olduğu açıktır. AlphaGo'nun en son çalışması AlphaGo Zero, yapay zekâ eğitiminin denetimli öğrenme adımlarını tamamen geride bırakmıştır ve AlphaGo Zero ekibine göre pekiştirmeli öğrenmeyle eğitilmiş bir sinir ağı kullanılarak kendi kendine oynamaya başlamıştır [317].

DeepMind, herhangi bir Atari 2600 oyununu oynayabilecek tek bir sinir ağı oluşturma hedefiyle pekiştirmeli öğrenme ve bir KSA aracılığıyla klasik oyunlarını oynamak için makineleri eğitmeyi başarmıştır [249]. Klasiklerden olan Atari 2600 oyunlarından yedisi, başlangıçta herhangi bir insanın sahip olduğundan daha hızlı tamamlanmıştır. İlk oyunlar başlangıçta mağlup edildiğinden, [318]'de tanımlanan Asenkron Avantaj Aktör Kritik (AAAK) algoritması gibi yeni algoritmalar, daha fazla oyunun tamamlanmasına veya çok daha kısa sürede çok yüksek bir ortalama puanının elde edilmesini sağlamıştır. AAAK algoritması, tüm sürecin verimliliğini en üst düzeye çıkarmak için eğitim sırasında farklı keşif politikaları kullanabilen tek bir makinede birden çok MİB kullanmıştır [318,319]. DeepMind [318]'deki çalışmada, farklı MİB iş parçacıkları kullanarak birden çok örnek çalıştırmanın, 16 çekirdekli bir MİB kullanmanın tamamen uygulanabilir olduğu, 48 MİB ve 8 GİB'ye ihtiyaç duyan birden fazla makinede önceki AlphaGo 2 adım yöntemine kıyasla çok daha ucuz olduğunu belirtmiştir [319].

OpenAI'nin DotA 2 ekibi, makinelerin aşırı düzeyde video oyunları oynamada ne kadar ileri gidebileceğini kanıtlamıştır. DotA 2, popüler Warcraft 3 modu Defence of the Ancients'in devamı niteliğindedir ve şu anda Valve Şirketi tarafından geliştirilmektedir. Bu oyun, birbirlerinin temel yapısını yok etmek amacıyla beş kişilik iki takımı karşı karşıya gelmesiyle oynanır. Oyun, 100'den fazla oynanabilir karakter ve satın alınabilecek yüzlerce oyun içi öge içermektedir. Bu durum, olası karakter ve öge oluşturma kombinasyonlarının neredeyse sonsuz olduğu anlamına gelmektedir. OpenAI botları, pekiştirmeli öğrenme kullanılarak ve birbirlerine karşı oyun oynadıkları A3C algoritması gibi yöntemler kullanılarak eğitilmektedir [320,321]. Tsukayama [321]'te, solo bir botun, e-sporun yıllık en büyük ödül havuzu turnuvası olan The International 7'de birebir oyunlarda profesyonelleri genel olarak kolaylıkla alt edebildiğini belirtmiştir. Bir sonraki adımda, amaç 5'e 5 oynanan oyunlarda galip gelmek olmuştur ve Quach'a göre, yetenek açısından yüzde 99.95'inde yer alan eski profesyoneller ve yarı profesyonellerden oluşan bir takıma karşı bunu yapmışlardır [322]. OpenAI Five, yolundaki tüm insan rakiplerini yok ederken, OpenAI, TI8'de rekabet eden profesyonellere, paiN Gaming ve Big God'a karşı iki gösteri maçı düzenlemeye karar vermiştir. OpenAI'nin etkinliğin sonuçlarıyla ilgili blog yazısında [323], rekabetin daha önce

karşılaştığı rakiplere kıyasla çok daha güçlü olduğunu, dolayısıyla her iki takıma karşı da kayıp olduğunu belirtmişlerdir. Bununla birlikte [323]'de oyunların hem şimdiye kadarki ortalamalarından daha uzun sürdüğü hem de çok iyi bir mücadele ortaya koyduğu ve herkesi etkilediği belirtilmiştir. [323]'de OpenAI, TI8'in sonuçlarından sonra sistemlerini geliştirmeye devam edecekleri ve DotA'nın içinde ve dışında daha iyi performans göstermek için yapay zekâyı geliştirmeye devam edecekleri belirtilmiştir. 25 yıl içinde yapay zekâ, dünyanın en iyi oyuncularını satrançta yenmekten dünyanın en büyük e-spor turnuvalarında yarışmaya kadar ilerlemeyi başarmıştır [316,317,320,322]. Video oyunlarında, sınır olmadığından yapay zekâ geliştirmek için uygun bir test alanı sağlanmaktadır.

4.2.2. Önerilen Yöntem

Pekiştirmeli öğrenmenin genel fikri, istenen veya istenmeyen bir eylemi gerçekleştirdikten sonra bir şeyi veya birini ödüllendirmek veya disipline etmektir. Bununla birlikte makine öğrenimi ile ilgili olarak, pekiştirmeli öğrenme; öğrenenin belirli bir durumda hangi eylemde bulunduğuna göre güçlendirildiği etkileşim yolunu tercih eden bir makine öğrenimidir [48,324]. Bu nedenle, makine belirli bir durumda yapılması mümkün olan en iyi eylemi öğrenmeyi amaçlamaktadır.

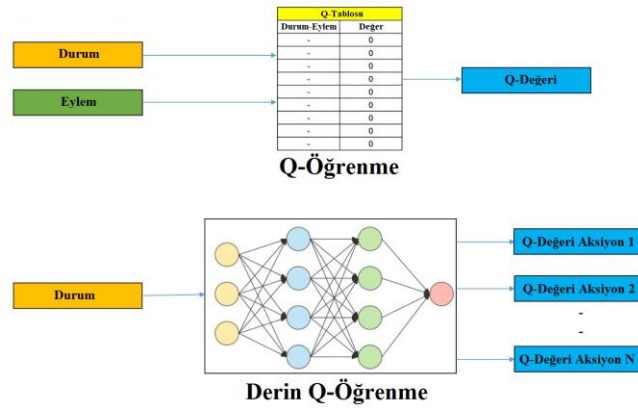
Pekiştirmeli öğrenme, kontrol (ör. Robotik, Otonom sürüş, vb.) ve karar verme için kullanılan algoritma ve teknikler ailesidir. Bu yaklaşımlar, MKS olarak ifade edilen sorunları çözmektedir. MKS; oyunun bir dizi durum S , bir dizi eylem A , bir ödül fonksiyonu R ve isteğe bağlı olarak durumlar arasındaki geçişleri tanımlayan bir geçiş fonksiyonu T ile tanımlanması gerektiği anlamına gelmektedir. Pekiştirmeli öğrenmeyi kullanmak için bu 4 bileşeni kullanarak problemin şekillendirilmesi gerekmektedir. Bu tez çalışmasında; geleneksel bir denetimli makine öğrenme yaklaşımı yerine DQÖ kullanılmıştır. DQÖ yaklaşımı, görünmeyen girdilere dayalı olarak hedeflerin nasıl tahmin edileceğini öğrenmeye çalışmaktadır. Bu çalışmada yapılan uygulamada, oyunun her durumunda yapılacak en iyi eylem bilinmediği için geleneksel bir yaklaşım etkili olmamaktadır. Pekiştirmeli öğrenme'de ortam (Pong oyunu) ve ajan (raket veya bu çalışmada raketin hareketlerini yönlendiren DSA) olmak üzere iki ana bileşen bulunmaktadır. Ajan bir eylemi her gerçekleştirdiğinde, ortam ajana, eylemin o belirli durumdan ne kadar iyi olduğuna bağlı olarak olumlu veya olumsuz olabilen bir ödül vermektedir. Ajanın amacı, her olası durum göz önüne alındığında, hangi eylemlerin ödülü en üst düzeye çıkardığını öğrenmektir. Durumlar, ajanın ortamdaki her iterasyonda elde ettiği gözlemlerdir. Bir durum, ajanın konumu, hızı veya ortamı tanımlayan herhangi bir değişken dizisi olabilir. Ajan tarafından eylemlerine karar vermek için kullanılan stratejiye politika denir. Teorik düzeyde bir politika, durum uzayından (ajan'ın alabileceği tüm olası gözlemlerin uzayı) eylem uzayına (ajan'ın gerçekleştirebileceği tüm eylemlerin uzayı, örneğin; yukarı, aşağı, sabit) yapılan bir planlamadır. Optimal ajan, her zaman olası en iyi eylemi tahmin etmek için ajanın daha önce hiç görmediği durumlar dâhil tüm durum

uzayı üzerinde genelleme yapabilir. Şekil 4.10'da Pekiştirmeli öğrenme ajanının ortam ile etkileşimi görülmektedir.

Q-öğrenme algoritmasında, her durumda eylem için maksimum beklenen gelecekteki ödülleri hesapladığı basit bir arama tablosu kullanılır. Temel olarak, bu tablo her durumda en iyi eyleme yönlendirecektir. Q-tablosunun boyutu, eylem ve durum değerlerine bağlıdır. Özellikle durum sayısı aşırı yüksek ise Q-öğrenme Q-tablosu oluşturulamadığı için uygulanamaz. Durum sayısı fazla olan ortamlar özellikle yüksek GİB'ler gibi donanımsal gereksinimi arttırabilmektedir. Diğer bir sorun ise eğitim ortamı ile test ortamı arasındaki durum farklılıklarıdır. Eğitim aşamasında eğitilen ajan, test aşamasında farklı bir durumla karşılaştığında yetersiz kalabilmektedir. Bu nedenle, bu tür problemler için Q-öğrenme tercih edilmez. Örneğin bir bilgisayar oyununda her piksel bir durum olarak değerlendirileceği için Q-tablosu ajanı yönlendirmede yetersiz kalacaktır. Bu sorunu çözmek için literatürde DSA ve Q-öğrenme bir araya getirilerek kullanılmıştır. Q-öğrenme ve DQÖ arasındaki karşılaştırma Şekil 4.11'de verilmiştir.



Şekil 4.10. Bir pekiştirmeli öğrenme ajanının çevre ile etkileşimi [20]

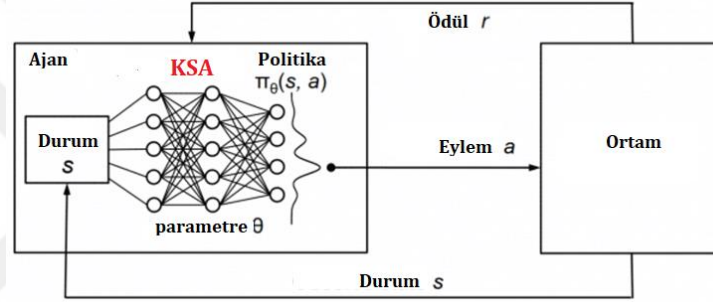


Şekil 4.11. Q-öğrenme ve DQÖ'nün karşılaştırılması

Video oyunlarında bir adım daha ileri gitmek için DQÖ yapısına konvolüsyonel işlem eklenir. Bu yeni mimari, literatüre Derin Konvolüsyonel Q-Öğrenme (DKQÖ) olarak geçmiştir [259]. Şekil 4.12'de, DKQÖ kullanırken pekiştirmeli öğrenme süreci gösterilmektedir. DQÖ algoritmasının sözde kodu Tablo 4.9'da gösterilmektedir.

Tablo 4.9. DQÖ algoritmasının sözde kodu

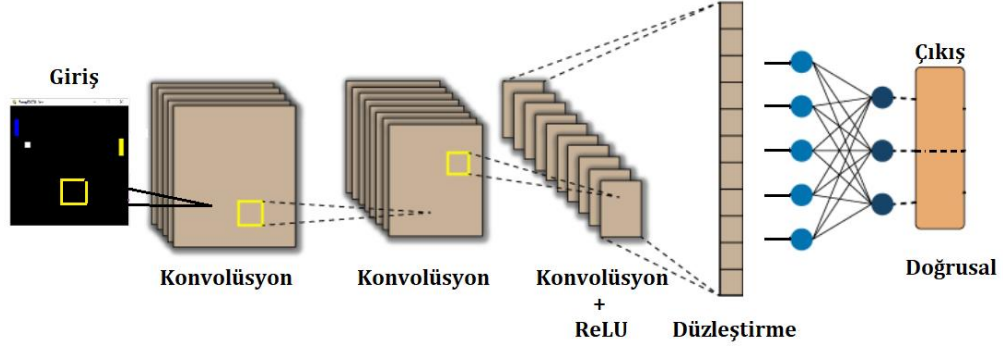
Replay Memory kapasitesini başlat
Ağı rastgele ağırlıkla başlat
Her bir episode için:
 Başlangıç durumunu başlat
 Her bir zaman adımı için:
 Bir eylem seç
 Keşif veya Sömürü yoluyla
 Seçili eylemi gerçekleştir
 Ödül ve sonraki durumu gözlemle
 Deneyimi ve replay memory’i kaydet
 Ağ ağırlıklarını güncelle



Şekil 4.12. DKQÖ ile pekiştirmeli öğrenme süreci [249]

DKQÖ ile yapılmak istenen, öğrenme aracısının video oyunundan kareleri girdi olarak kullanmasını sağlamak ve bu şekilde oyunun ajan tarafından nasıl oynanacağını öğrenmektir. KSA, görüntüyü önceden işlemek ve standart sinir ağının çalışabileceği bir forma sokmak için kullanılır. Bunu yapmanın ilk adımı, giriş görüntüsündeki belirli özellikleri veya nitelikleri tespit etmektir. Bu işlem, konvolüsyonel katman tarafından yapılır. Bu katman, kenarlar ve eğriler gibi düşük seviyeli özelliklerin yanı sıra yüz veya el gibi daha yüksek seviyeli özellikleri tespit etmek için filtreler kullanır. Ardından KSA, görüntüden doğrusallığı kaldırmak için ek katmanlar kullanır. Doğrusallık kaldırıldığında, görüntüyü sıkıştırmak ve verileri düzleştirmek için ek katmanlar kullanılır. Son olarak, bu bilgi, KSA dünyasında tam bağlı katman adı verilen bir sinir ağına iletilir. Basit bir KSA yapısı Şekil 4.13’te gösterilmektedir.

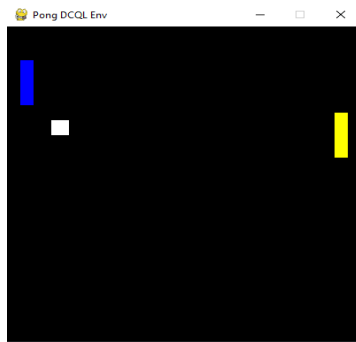
Bu uygulama çalışmasında kullanılan KSA yapısında 3 Konvolüsyon, 1 düzleştirme katmanı, 1 yoğun katman ve 1 doğrusal katman olmak üzere 6 katmanlı bir model önerilmiştir.



Şekil 4.13. Pong oyunu uygulaması için önerilen KSA yapısı

4.2.3. Uygulama Sonuçları

Bu tez çalışmasında, pong oyunu için Pygame kütüphanesi kullanılarak pekiştirmeli öğrenme ortamı oluşturulmuştur. Pygame, video oyunları yazmak için tasarlanmış bir dizi Python modülüdür. Pygame, Basit DirectMedia Katmanı anlamına gelen SDL kitaplığını kullanır. SDL, sistemin ses, video, fare ve klavye gibi temel multimedya donanım bileşenlerine platformlar arasında erişim sağlar. Bu durum, python dilinde tam özellikli oyunlar ve multimedya programları oluşturulmasına olanak tanır. Şekil 4.14'te görüldüğü gibi oluşturulan pekiştirmeli öğrenme ortamında iki raket ve bir top bulunmaktadır. Soldaki mavi raket DKQÖ tabanlı bir yaklaşımla hareket etmeyi öğrenmektedir. Sağdaki mavi raket ise herhangi bir yapay zekâ yaklaşımının kullanılmadığı tamamen matematiksel olarak hareket etmektedir. Beyaz top rakipten döndükten sonra her iki raket yukarı ya da aşağı doğru yönelebilmektedir. DKQÖ ile eğitilen ajanın (mavi raketin) amacı gelen topu tekrar sarı rakete doğru yollamaktır. Ajan aşağı, yukarı ve sabit olmak üzere üç farklı aksiyon gerçekleştirmektedir. Ajan aksiyon seçimini yaparken, politikasını ϵ -Greedy'ye göre seçer. ϵ -Greedy politikası, ajanın her yinelemede sabit bir değere sahip bir eylem seçmesine olanak tanımaktadır. Aracın ortamı daha iyi öğrenmesi için başlangıçta seçilen ϵ değeri her yinelemede belirli oranda azaltılır. Ajan, ilk iterasyonlarda büyük ϵ değerinden dolayı genellikle ortamı keşfeder (exploration), sonraki iterasyonlarda ise deneyim sahibi olacağı için sömürü (exploitation) yöntemine başvurur. Bu çalışmada ϵ -Greedy politikası için benzer yaklaşım kullanılmıştır.



Şekil 4.14. Pygame kullanılarak oluşturulan Pong Oyunu ekranı

Bu tezde yer alan DPÖ modelinin implementasyonunda, Keras kütüphanesi kullanılmıştır. Önerilen KSA modelinde; 3 Konvolüsyon, 1 düzleştirme katmanı, 1 yoğun katman ve 1 doğrusal katman olmak üzere 6 katmanlı bir yapı kullanılmıştır. Son olarak çıkış katmanında aksiyon sayısı (3) kadar nörondan oluşan bir yoğun katman kullanılmıştır. Konvolüsyon adımlarında sırayla filtre sayısı 32, 64 ve 128 olarak alınmıştır. Kernel boyutu ise tamamında 2X2 ve aktivasyon fonksiyonu ReLU olarak seçilmiştir. DKQÖ algoritmasının eğitim aşamasında kullanılan parametreler ve hiper-parametreler Tablo 4.10'da gösterilmektedir.

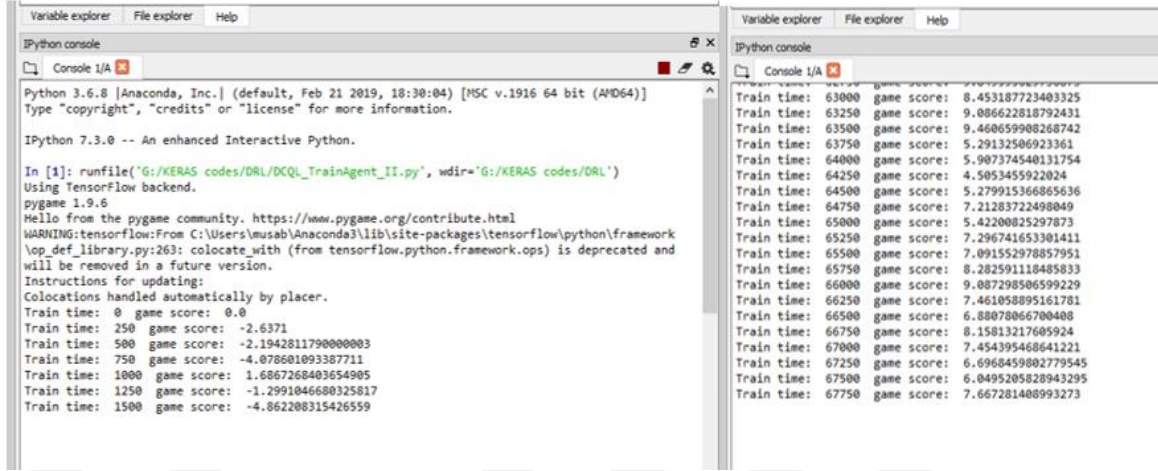
Tablo 4.10. DKQÖ algoritmasının eğitim aşamasında kullanılan parametreler ve hiperparametreler

Hiper-Parametre	Değer	
Eğitim Adım Sayısı	10000	
Öğrenme Oranı	Adaptif	
Gamma	0,975	
Keşif Oranı	Maksimum Keşif	1
	Minimum Keşif	0.05
	Azalma Oranı	0.95
Batch Boyutu	64	

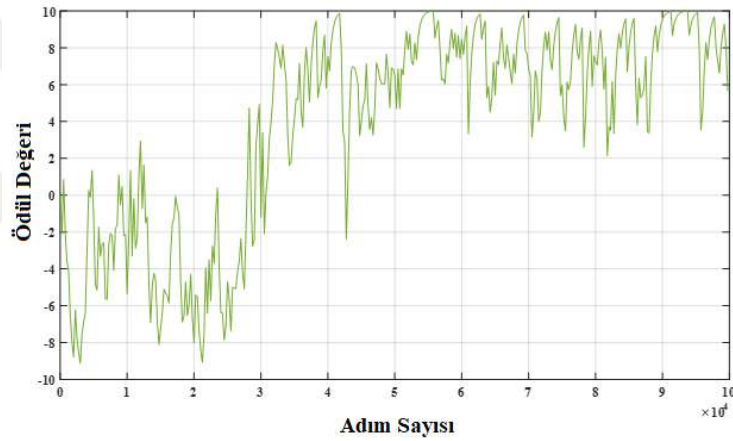
Şekil 4.15'te eğitim sürecinde elde edilen skor değerinin değişimi verilmiştir. OKH, kayıp fonksiyonu olarak kullanılmıştır. Optimizer olarak Adam seçilmiştir. Adam, eğitim verilerinde yinelemeli olarak ağırlıklarını güncellemek için klasik stokastik gradyan iniş prosedürü yerine kullanılabilen bir optimizasyon algoritmasıdır. Stokastik gradyan inişi, tüm ağırlık güncellemeleri için tek bir öğrenme oranı sağlar ve eğitim sırasında bu öğrenme oranı değişmez. Adam ile öğrenme oranı değişmektedir. Çıkış katmanında aktivasyon fonksiyonu olarak lineer fonksiyon kullanılmıştır. Şekil 4.16'da ödül-adım sayısı grafiği elde edilmiştir. Grafiğe bakıldığında, DKQÖ modeli ile yapılan eğitim sırasında, başlangıçta elde edilen ortalama ödül değerlerinin eğitim aşamasının sonuna doğru arttığı görülmektedir. Bu da DKQÖ tabanlı eğitilen ajanın Pong oyununu oynamayı öğrendiğini göstermektedir.

Bu uygulamanın önemli yönlerinden biri de DQA algoritmasında klasik KSA mimarilerinde kullanılan örnekleme işleminin yapılmadığıdır. Örnekleme işlemi, nesne algılama, görüntü sınıflandırma vb. gibi, nesnenin görüntüdeki konumunun dikkate alınmadığı ve yalnızca görüntüde istenen nesnenin bulunup bulunmadığı bilgisi istendiğinde önemlidir. Örneğin, bir resimde köpek olup olmadığının belirlenmesi isteniyorsa, sadece resimde köpek olup olmadığına bakılır ve resimdeki köpeğin konumu kontrol edilir. Bu nedenle, köpeğin konumundan bağımsız olarak görüntüde bir köpek olup olmadığını belirlemek için bir örnekleme işlemi kullanılır. Ancak DQA'da örnekleme işlemi yapılmamalıdır çünkü mevcut oyun durumunu anlamak için pozisyon oldukça önemlidir. Örneğin, Pong'da amaç oyun ekranında bir top olup olmadığını sınıflandırmak

değildir. Önemli olan daha iyi bir eylem gerçekleştirmek için topun konum bilgisinin elde edilmesidir. Bu sebeple örnekleme işlemi DKQÖ mimarisinde kullanılmamaktadır.



Şekil 4.15. Eğitim aşamasında elde edilen skor değerinin değişimi



Şekil 4.16. Uygulama sonunda elde edilen ödül değerinin değişimi

4.2.4. Sonuçlar

Bu çalışmada, herhangi bir veri kümesi kullanılmadan pekiştirmeli öğrenme algoritmalarında temel ilke olan kendi kendine öğrenme politikası kullanılmıştır. Bir yapay zekâ ajanının Pong oyunu oynamayı öğrenmesi için uygun bir DPÖ modeli geliştirilmiştir. Ajan girdi olarak yalnızca ham pikselleri kullanarak öğrenme yeteneğini iyileştirmiştir.

4.3. Düello DQA Algoritması ile Robotik Kavrama Uygulaması

Hızla ilerleyen teknoloji sayesinde, robot kullanımı hayatımızın birçok alanında her geçen gün daha fazla yer almaktadır. Tasarlanan farklı türdeki robotlar belirli alanlarda gereken insan gücünün yerini alarak veya hafifleterek insan yaşamının kalitesini arttırmayı hedeflemektedir. Günümüzde robot manipulatörler; yalnızca yeni ve gelişmekte olan bir araştırma alanı olarak değil

aynı zamanda birçok amaç için uygulanabilir ve kullanışlı bir araç olarak bilim adamları, mühendisler ve hatta klinisyenler için en önemli ilgi alanlarından biri haline gelmektedir. Farklı durumlara ve uygulamalara bağlı olarak manipülasyon ve kavrama için çok çeşitli tanımlar vardır. Genel olarak mekanik manipülasyon; bir nesne üzerinde harekete veya deformasyona neden olan bir kuvvet veya tork uygulama eylemi olarak tanımlanabilirken bir nesneyi tutmak, kavrama olarak adlandırılır. Genel olarak manipülatörlerin farklı uygulama türleri vardır. Manipülatörler tarafından gerçekleştirilen yaygın prosedürlerden biri kavramadır. Bir kavrama görevinin gerçekleştirilmesi için, her manipülatör, manipülatörün uç işlevcisine monte edilmiş bir kavrayıcıya ihtiyaç duyar. Bu tez çalışması kapsamında robotikte karşılaşılan önemli problemlerden biri olan kavrama görevinin uygulamasına yönelik robotik bir kavrama görevinin simülasyonda ortam tasarımı ve gerçekleştirilmesi amaçlanmaktadır.

4.3.1. İlgili Literatür

Robotlara algılama becerisi kazandırmak, robotik ve bilgisayarla görü uygulamalarında köklü bir hedef haline gelmiştir. Bir robotu zeki olarak tanımlamak için robotun çevreyi gözlemlemesi, algılaması ve onunla etkileşime girmesi gerekmektedir. Kavrama, robotlarda olması gereken temel yeteneklerinden biridir ve oldukça önem taşımaktadır. Örneğin; ev robotları yaşlı veya engellilerin günlük kavrama görevlerinde yardımcı olabilecek potansiyele sahiptir. Robotik kavrama alanında birçok araştırma yapılmıştır [5,17,18,107,108,325–328]. Bu tez çalışmasında daha ziyade son on yılda yapılan güncel yöntemler üzerinde durulmuştur. Önerilen yöntemi öne çıkaran özellik, DPÖ'ye dayalı uçtan uca bir kavrama stratejisidir. Bu nedenle, kavramada karşılaşılan sorunları çözmek için literatürdeki diğer yöntemlere kısaca değinilecektir.

Kavrama öğrenme stratejilerini geliştirmek için farklı yaklaşımlar çalışılmıştır. Öğrenme temelli yaklaşımlar, veri üretmek için gerçek donanım kullanan yöntemlerden simülatörlerde yapay eğitim verileri oluşturan yöntemlere kadar çeşitlilik gösterir [7]. Bir robotun kavraması, genellikle hedef nesnelerin görüntüleriyle planlanır. Ciocarlie, makine öğrenimi tekniklerini ve 3B nokta bulut verilerinden geometrik poz tahmini içeren bir yöntem kullanmıştır. Bu yöntem, ayarlanması zor olan birçok hiperparametreye sahip olabilir [329]. Kavramaya yönelik bir başka yaklaşımda, 3B nesne pozunu ve şeklini doğrudan derinlik ve renkli görüntülerden [5,17,107,330] tahmin etmek için derin öğrenme kullanılmıştır. Diğer bir yaklaşımda ise bir kavrama gerçekleştirmek için nesneleri ve pozlarını her zaman net bir şekilde tanımak gerekmeyebileceğinden, nesne pozunu ve şeklini açıkça temsil etmeden görüntülerdeki kavranabilir bölgeleri doğrudan tespit edilmiştir [16,108].

Pinto ve Gupta, bir görüntüde ilgilenilen bölgeyi belirlemek için bir arka plan azaltma yaklaşımı kullanmıştır ve bir girdi görüntü yaması verildiğinde farklı kavrama yönleri için kavrama olasılığını tahmin eden KSA tabanlı bir sınıflandırıcıyı eğitmişlerdir [16]. Sayısız deneme yapılma

işlemi gerçekleştirmeyi hesaba katan eksiksiz bir kendi kendini denetleyen aşamalı müfredat öğrenme sistemini tanıtmışlardır. Chu ve arkadaşları, tek bir nesne ile senaryolarda kavrama algılamaya odaklanmıştır. Oryantasyon regresyonunu, bir sınıflandırma zorluğuna dönüştüren ve tek bir sonucun tercihi için çok sayıda aday kavramayı öngören bir mimari sunmuşlardır [331]. Mahler ve arkadaşları, başarı oranını anında tahmin etmek için görüntü uzayında tek tip ve rastgele bir aday kavrama kümesi seçmek ve bir kavrama kalitesi KSA mimarisi kullanarak her aday kavramayı değerlendirmek için bir teknik önermiştir [10]. Zhang ve arkadaşları, iki aşamadan oluşan bir robotik kavrama algılama algoritması sunmuşlardır. Çalışmalarında ilk aşamada giriş görüntüsündeki ilgilenilen bölge algılanmıştır ve ikinci aşamada bu bölgenin özelliklerine dayalı bir kavrama dedektörü kullanılmıştır [332]. Zeng ve arkadaşları, kendi kendini denetleyen DPÖ yöntemini kullanarak tam bir KSA mimarisini eğitip eğitilmiş politikaları yeni nesnelere genellemeyi göstermişlerdir [326]. Jang ve arkadaşları tarafından geliştirilen bir yöntemle göre kavranan nesne, kavramadan önceki ve sonraki sahne arasındaki özellik farkıyla kabaca tanımlanabilir [333]. Deng ve arkadaşları ise kendi kendini denetleyen 6B nesne poz tahmini için bir robotik sistem önermiştir [334]. Robotik kavrama için Moussa ve arkadaşları, bir nesneyi kavramak için insan eylemine bağlı olarak öğrenme stratejisinin değiştirildiği önemli çalışmaları derlemişlerdir [335].

Caldera ve arkadaşları, geliştirilmiş robotik kavramaya derin öğrenme yöntemleri uygulamak için en son teknoloji üzerine bir çalışma yapmışlardır. Yaptıkları çalışmada, derin öğrenme metodolojisinin her bir bileşeninin robotik kavrama algılamasının genel performansını nasıl geliştirdiğini incelemişlerdir [105]. Derin öğrenme, çok büyük boyutlu durumlar ve eylem alanları gibi önceden zor karar verme senaryolarına uygulanabilmesi için pekiştirmeli öğrenmenin de gelişmesine önemli katkılar sağlamıştır. DQA'lar ve varyantları üzerine yapılan son çalışmalarda, DPÖ'nün kontrollü kavrama stratejileri tasarlamak için nasıl uygulanabileceğini gösterilmiştir [30,328,336,337]. Öte yandan robotik kavramadaki zorlukların üstesinden gelmek için Güvenli Bölge Politika Optimizasyonu algoritması, Derin Deterministik Politika Gradyan Algoritması ve Aktör Kritik algoritmaları gibi başka DPÖ yöntemleri de bulunmaktadır [336,338]. Ancak bu tez çalışmasında; DPÖ'nün simülâtörde ne kadar iyi performans gösterebileceğini gözlemlemek için bir DQA varyantı kullanılmıştır.

4.3.2. Uygulama Organizasyonu

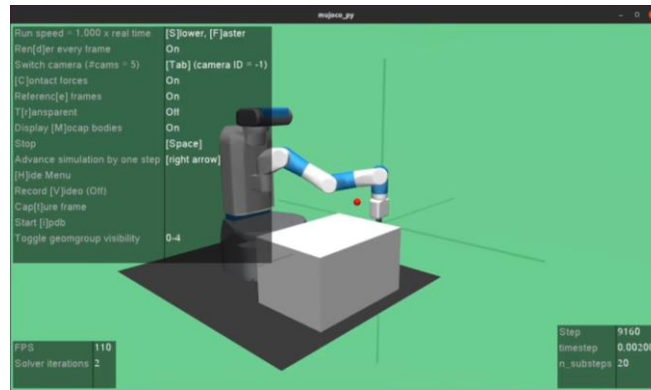
Düello DQA algoritması ile robotik kavrama uygulamasında ilk olarak, simülâtör tercihi yapılmıştır. Daha sonra robotik kol olarak seçilen KUKA youBot yapısı incelenmiştir. Ardından Webots'ta ortam bileşenleri olan robotik kol, kavranması gereken nesne ve Kinect kamera bir araya getirilerek uygulama düzeneği oluşturulmuştur, sonra kavrama eyleminin gerçekleştirilmesi için gerekli yörünge nasıl oluşturulacağı belirlenmiştir. Yörünge planlaması yapıldıktan sonra

kontrolör yardımıyla, düello DQA algoritması tabanlı kavrama eğitimi gerçekleştirilmiştir. Böylece robotik kavrama süreci tamamlanmıştır. Bu bölümde ayrıca uygulama ortamı, ödül ve durum yapıları anlatılmıştır.

Simülasyon Seçimi

Simülasyon seçimi, robotiğin temellerini oluşturduğu ve tüm yazılım mimarisini kapsadığı için önem arz etmektedir. Özellikle sürekli kontrol optimizasyon görevleri için simülasyonun hatasız olması, robotik kontrolörün performansında önemli bir rol oynar [339]. Her simülasyonun avantajları ve dezavantajları bulunmaktadır. Bu tez çalışmasında, güncel simülasyonlardan Mujoco, Gazebo, Pybullet ve Webots incelendikten sonra Webots'un kullanılmasına karar verilmiştir. Webots; hızlı destek, topluluk, istikrar, ölçeklenebilirlik ve kullanım kolaylığı gibi öne çıkan özelliklere sahip olması sebebiyle seçilmiştir.

2015 yılında piyasaya sürülen MuJoCo (Multi-Joint Dynamics with Contact), çoklu eklem kontrolündeki kelimelerin İngilizcesinin ilk iki harfi kullanılarak isimlendirilmiştir [22]. C dilinde geliştirilen Şekil 4.17'de görüldüğü gibi bir uygulama programlama ara yüzüne (API) sahiptir. C dilindeki API, donanımı etkin ve hızlı bir şekilde kullanmak için araştırmacılara büyük bir avantaj sağlamaktadır. Ayrıca simülasyonu kolay bir şekilde kontrol etmek için python API'si sunulmaktadır. MuJoCo-py diye adlandırılan bu API, OpenAI ürünü olup MIT lisansı altında tamamen açık kaynaktır.



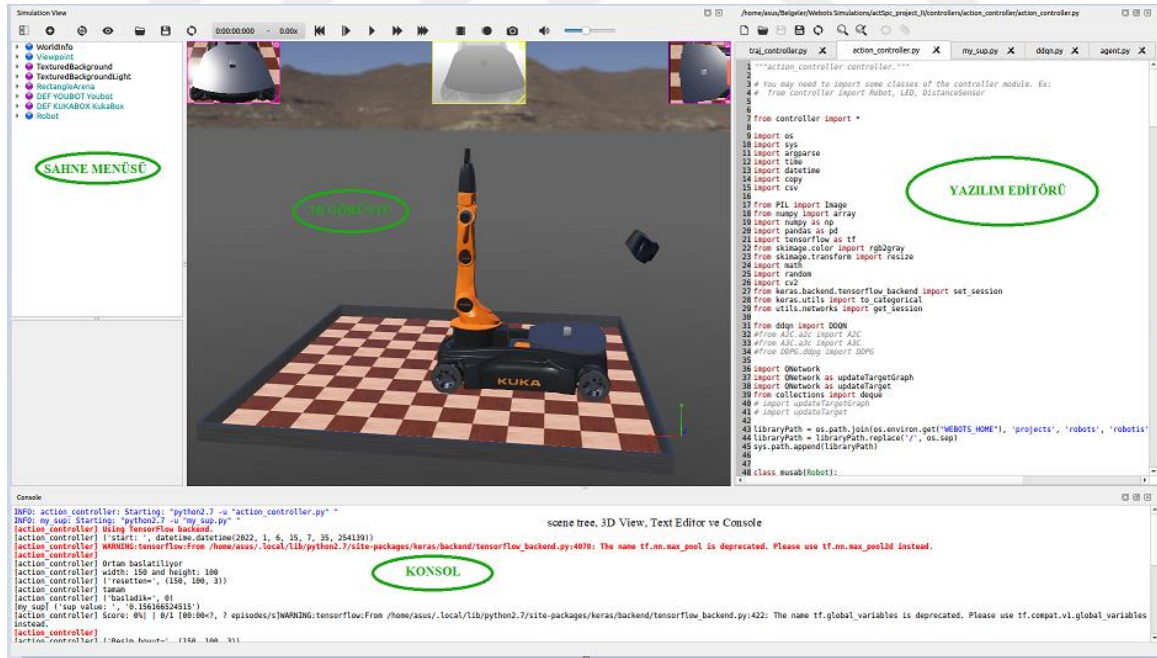
Şekil 4.17. Örnek MuJoCo Ortamı

Diğer taraftan araştırma amacıyla kullanımda MuJoCo lisans gerektirmektedir. Lisans maliyetinin olması MuJoCo kullanımının olumsuz yönlerinden biridir. Bir başka faktör ise ihtiyaç duyulan eğitim süresinin yüksek olmasıdır. Dolayısıyla MİB kullanan MuJoCo için GİB ihtiyacı ortaya çıkmaktadır.

Gazebo'nun geliştirilmesine, 2002'de Güney Kaliforniya Üniversitesi'nde başlanmıştır [23]. Sonraki yıllarda Willow Garage devraldıktan sonra Gazebo'yu ROS ve PR2'ye entegre etmiştir. Böylece Gazebo ROS topluluğunun birincil simülasyonu haline gelmiştir ve 2012 yılında OSRF'nin (Açık Kaynak Robotik Vakfı) bir parçası olmuştur. Teknik olarak Gazebo, Bullet, ODE,

yazılımları, ticari olarak mevcut olan mobil robotlara transfer edilebilmektedir. Webots ile bir veya daha fazla, benzer veya farklı robotların aynı ortamı paylaşabildiği 3 boyutlu ortamlar oluşturulabilmektedir. Ortama eklenen her bir nesneye ait şekil, renk, doku, kütle gibi belirli sayıda özellikler kullanıcı tarafından belirlenip değiştirilebilmektedir. Ortamda kullanılan robotlara istenilen sayıda sensör ve aktüatör eklenebilmektedir. Windows, Mac ve Linux işletim sistemleri altında çalışabilmektedir. Webots, robot kontrolörlerinin tasarlanmasını MATLAB, C, C++, Java ve Python programlama dillerinde oluşturma imkânı sağlamaktadır. Kullanılan diller kendi editöründe derlenebilmektedir. Ara yüzü basit ve kullanışlıdır. Webots Simülâtörü, Lausanne şehrinde bulunan İsviçre Federal Teknoloji Enstitüsü ile işbirliği yapılarak Dr. Oliver Michel tarafından geliştirilmiştir. Test aşamaları ve doküman hazırlığı 7 yıl sürmüştür.

Bu tez çalışmasında, Şekil 4.20’de görülen Webots simülâtörünün R2019a versiyonu kullanılmıştır. Kullanılan simülâtör, içerdiği kütüphaneler sayesinde farklı fiziksel çevre modelleri, robot ve robot kollarını barındırmaktadır. Endüstriyel silahlar, insansı robotlar, uzay araçları, çok ayaklı robotlar, akıllı mobil robotlar, arabalar, insansız araçlar, otonom su altı araçları ve her türlü robot modellenmektedir. Hâlihazırda Adept, Bluebotics, Boston Dynamics, Clearpath, Epfl, Fujitsu, Gctronic, iRobot, K-team, Kondo, Kuka, Lego, Micromagic, Mobsya, Nasa, Neuronics, Nex, Parallax, Robotis, Sacon, Softbank, Sony, Sphero, Surveyor, Tinkerbots, Unimation şirketlerine ait robot platformları bulunmaktadır.



Şekil 4.20. Webots ortamı

Robotik görevlerin gerçek robotta denenmeden önce simülasyonda yapılmasının daha kısa sürede olmaktadır. Maliyetinin düşük olması ve olası hatalarda herhangi bir maddi kaybın yaşanmaması gibi birtakım olumlu yönleri bulunmaktadır. Ayrıca kullanıcıya daha iyi tasarım

imkanları sunulmakla birlikte simülasyon ortamındaki robotlarla eğitilen modellerin gerçek robotlara aktarılmasını sağlamaktadır.

Webots simülatörü Şekil 4.20’de görüldüğü gibi sahne menüsü (scene tree), 3B görüntü (3D View), yazılım editörü (Text Editor) ve konsol (Console) olmak üzere dört temel alandan oluşmaktadır. Sahne menüsü ile Webots dünyasında kullanılan nesnelere ait bazı özelliklerin değiştirilebilme imkânı sunulmaktadır. 3B görüntü ile üç boyutlu dünya görsel olarak takip edilmektedir. Yazılım editörü ile kontrolör yazılımları tasarlanmaktadır. Konsol ile de yazılımdaki hatalar tespit edilip ayıklanmakta ve simülasyondaki süreç izlenmektedir. Webots simülatörü ile ilgili bilinmesi gereken temel yapılar dünya, kontrolör ve denetmendir.

Dünya (World): Robotun ve robotun bulunduğu ortamın özelliklerinin üç boyutlu tanımıdır. Pozisyon, oryantasyon, geometri, görünüm (renk ya da parlaklık gibi), fiziksel özellikler, nesne tipi gibi her bir nesneye ait tanımlamaları içermektedir. Dünyalar, nesnelerin başka nesneleri içerdiği hiyerarşik bir yapıdan oluşmuştur. Örneğin dünyada oluşturulan bir robot iki tane tekerlek, bir tane mesafe sensörü ve üzerinde kamera bulunduran bir tane servo yapıdan oluşabilir. Dünya dosyası, robotların kontrol kodunu içermemektedir. Sadece her bir robot için gerekli olan kontrolörün ismini belirtmektedir. Dünyalar .wbt uzantılı olarak kaydedilmektedir. .wbt uzantılı dosyalar her bir simülatör projesinin dünyalar alt dizininde kaydedilmektedir.

Kontrolör: Bir dünya dosyasında belirtilen bir robotu kontrol eden bilgisayar yazılımıdır. Kontrolörler, simülatör tarafından desteklenen programlama dillerinden biri ile yazılabilmektedir. Simülasyon başlatıldığında simülatör belirtilen kontrolörlerin her birini ayrı bir işlem gibi başlatmaktadır. Aynı kontrolör kodu bir kaç robot tarafından kullanılabilir. Ancak her bir robot için ayrı bir işlem başlatılmaktadır. Bazı programlama dillerinin derlenmesi, bazılarının yorumlanması, bazılarının ise hem derlenmesi hem de yorumlanması gerekmektedir.

Denetmen (Supervisor): Gerçek bir robot tarafından gerçekleştirilmesi güç olan insan tarafından gerçekleştirilebilen işlemleri yürütebilen özel tip bir robottur. Denetmen, bahsedilen programlama dillerinden biriyle yazılabilen bir kontrolör programı ile ilişkilendirilmektedir. Normal kontrolöre göre denetmen kontrolör, özel işlemleri gerçekleştirebilmektedir. Bu özel işlemler robotun rasgele bir pozisyona hareket ettirilmesi, simülasyonun kaydedilmesi benzeri simülasyon kontrollerini içermektedir.

Robotlara eklemek için mesafe, kızılötesi, ultrasonik, dokunmatik, pozisyon, enkoder ve kamera gibi sensörler, sensör kütüphanesinde bulunur. Benzer şekilde aktüatör kütüphanesi bulunmaktadır. Aktüatör kütüphanesinde diferansiyel tekerlek motor ünitesi, bağımsız tekerlek motorları, servo motorlar, LED’ler, alıcı-verici üniteleri ve kavrayıcılar bulunmaktadır.

Webots’ta mobil robot simülasyonları için OpenGL ve açık dinamik makinası (Open Dynamics Engine) teknolojisi kullanılarak kompleks ortamlar oluşturulabilmektedir. Simülasyon ortamı PNG veya JPEG formatında ve MP4 formatında kaydedilebilmektedir. VRML97 standardı

ile 3-boyutlu modellemenin yapıldığı yazılımlardaki nesneler dışarıdan alınabilmektedir. Webots'ta oluşturulan dünyanın boyutuna bakılmaksızın simülasyon hızlı bir şekilde yapılması için optimize edilmektedir. Kompleks robotlar servo düğümler bir araya getirilerek oluşturulmaktadır. Bu da istenilen sayıda ekleme sahip robotlar, robot kolları ve kamera sistemlerinin oluşturulmasına olanak sağlar.

Webots'ta kullanılan simülasyon sistemi, sanal bir zaman birimini kullanmaktadır. Dolayısıyla simülasyon gerçek dünyadaki bir robota göre daha hızlı gerçekleşebilmektedir. Webots'ta zaman adımları doğruluk ya da hız ihtiyacına göre belirlenmektedir. Robotun nasıl davrandığını anlamak için adım-adım tekniği de kullanılabilir.

Eklemler mekanik parçalar içeren kompleks robotların simülasyonu gerçekleştirmek için tam bir fiziksel simülasyon gerektirir. Robotlara ait her bir bileşen için kütle dağılım matrisi, statik ve dinamik sürtünme katsayıları tanımlanmaktadır. Çarpışmayı önlemek için her bir bileşen sınırlara sahip bir nesne ile ilişkilendirilmektedir. Servo sistemlere ait hız, tork ve pozisyon gibi büyüklükler yazılım ile kontrol edilebilmektedir. Simülasyonda testleri tamamlanmış kontrolör yazılımlar, çapraz-derleme ile gerçek robotlara aktarılabilir.

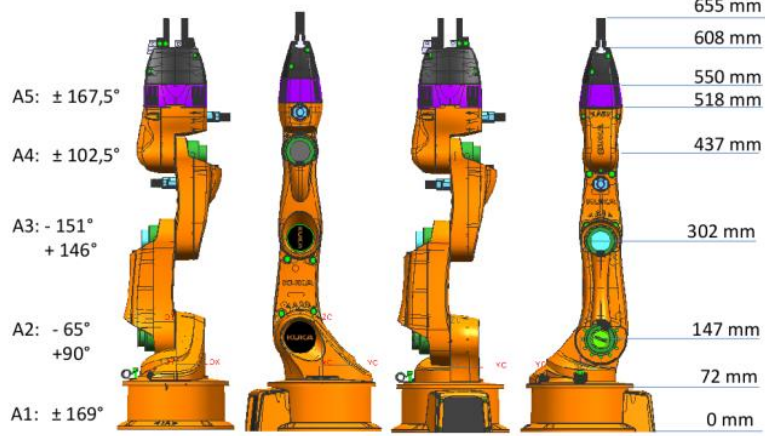
Simülasyonda Kullanılan Robotik Manipülatör

Bu tez çalışmasında nesnelerin kavranması için gelecekte endüstride yaygın olarak kullanılması beklenen ve Webots'ta hazır olarak gelen KUKA youBot robotu kullanılacaktır. KUKA youBot robotu, eğitim ve araştırma amaçlı kullanılabilir. YouBot hareketli bir platform ve 5 eksenli bir koldan oluşmaktadır. Ayrıca youBot kolu iki parmaklı bir kavrayıcıya sahiptir. Hareketli platformu sayesinde istediği yönde hareket edebilmektedir. YouBot'un tabanı ile kolu arasındaki bağlantı EtherCAT denilen bir bağlantı sağlanmaktadır.

KUKA Şekil 4.21'de görüldüğü gibi youBot kolu 5 dönel eklemden ve uç işlevci olarak kullanılan iki parmaklı bir kavrayıcıdan oluşmaktadır. YouBot kolu 6.3 kg olup 655 mm yüksekliğe ulaşabilmekte ve 0.5 kg ağırlık taşıyabilmektedir. YouBot kolunun dönel eklemleri iki farklı eksenle hareket edebilmektedir. A2, A3 ve A4 eksenleri paralel olarak aynı eksen etrafında dönmektedir. A1 ve A5 eksenleri ise youBot kolu yukarıya doğru düz bir şekilde yönlendirildiğinde paralel olarak aynı eksen etrafında dönmektedir. Eklemlerden her birinin maksimum hızı saniyede 90 derecedir. Uç işlevcisi parmak başına 11.5 mm açılabilir. Şekil 4.22'de KUKA youBot robot kolunun yükseklik ölçüleri ve açı sınırları verilmiştir.



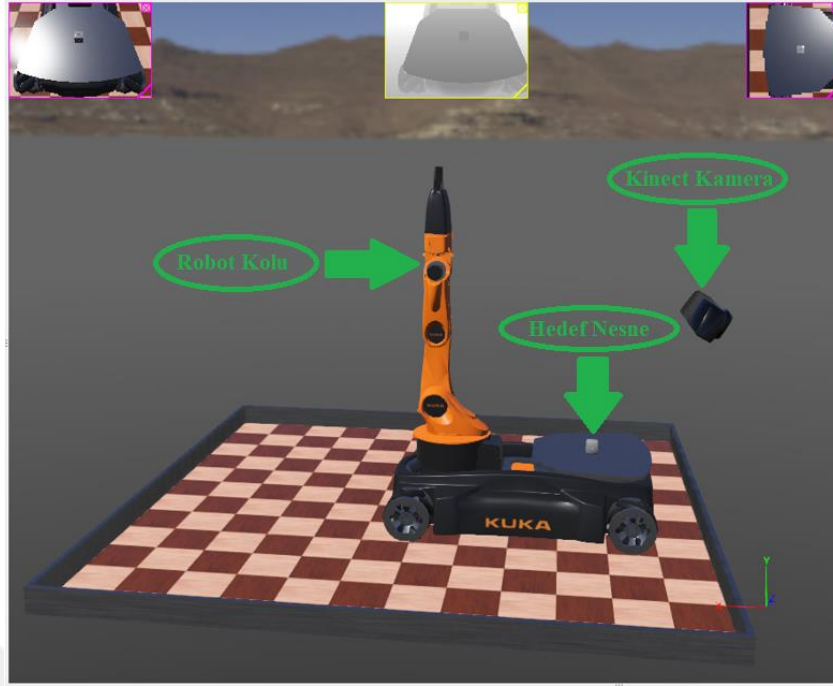
Şekil 4.21. Simülasyonda kullanılan 5 serbestlik dereceli KUKA youBot robot kolu yapısı



Şekil 4.22. KUKA youBot robot kolunun yükseklik ölçüleri ve açı sınırları

Webots'ta Oluşturulan Deney Düzeneği

Deney düzeneği Şekil 4.23'te görüldüğü gibi oluşturulmuştur. Robot kolu olarak kullanılan KUKA youBot'un tabanı, deney sürecinde sabit tutulup hareket ettirilmemektedir. Robot kolu, 5 eksenli döner bir yapıya sahip olup uç işlevci olarak ikili bir kavrayıcıya sahiptir. İkili kavrayıcıya, paralel kavrayıcı da denmektedir. Hedef nesne olarak kübik yapıda bir nesne ortama eklenmiştir. Kinect kamera robotik yapıdan fiziksel açıdan bağımsız olarak farklı bir noktaya yerleştirilmiştir. Kinect kameradan alınan görüntüler Şekil 4.23'te sol üstte görülmektedir.



Şekil 4.23. Deney Düzenneği

Robot Kolu Hareketi için Yörünge Planlaması

Bir robot kolunun, ortamda başlangıç pozisyonu ve yönelimi ile hedef pozisyonu ve yönelimi verildiğinde, uç işlevcinin başlangıçtan hedefe doğru sorunsuz bir yörüngede hareket etmesi istenir. Bu işlem, robot kolun eklem hızlarını ve pozisyon değerleri hesaplanarak yapılabilir. Bu doğrultuda kullanılan robot kolunun yörünge planlaması yapılır. Yörünge planlamasının yapılmaması durumunda robot kolun belirlenen hedef pozisyon konumuna nasıl bir hareket sonucunda geleceği tam olarak bilinmemekle birlikte, robot kol bu hareketini gerçekleştirirken ortamda bulunan bir cisme veya zemine çarpabilir. Bu sebeple eklem uzayında veya kartezyen uzayda yörünge planlaması yapılır.

Yörünge değerlerinin elde edilmesi için Webots kullanıcı kılavuzundan eklem değerleri incelenmiştir. Kavrayıcı parmakları doğrusal değişen bir yapıya sahip iken eklemler döner yapıya sahiptir. Açığa bağlı olarak eklemlerin farklı pozisyon değerleri üretilmektedir. Klavyeden manuel kontrol ile eklemler hareket ettirilerek, kavramaya yönelik her bir ekleme ait pozisyon değişimleri elde edilmiştir. Ortam denetmenine aktarılan bu değerler, kavram süreci başladığında robot kolu kavram eylemlerini gerçekleştirmek için uygun pozisyona getirilmiştir.

Problemin Tanımlanması

Simülasyon ortamında, ikili kavrayıcı kübik yapıda bir nesneyi kavramaya çalışır. Kavrayıcı, bir kol yapısına ve tabana sahiptir. Herhangi bir kinematik hesaplaması burada yok sayılır. Gözlenen ortam durumu, ortama eklenmiş Kinect kameradan alınır. Kavrayıcı, sürekli

görüntü girişi ile kavrayıcı parmaklarına konum kontrolü sağlar. Öğrenme süreci, belirli bir zaman adımından sonra sonlandırma sinyali ile biter. Kavrayıcı parmakları, Düello DQA algoritması tarafından tanımlanan önceden tanımlanmış konumlara ulaştığında başarı sinyali tetiklenir. Simülasyonda, KUKA youBot plakasına bırakılan bir kutuyu kavramaya çalışmaktadır.

Robotik Ortamının Tanımlanması

Webots simülatöründe oluşturulan test ortamı, algoritmaları hızlı bir şekilde test etmek için kullanılmıştır. KUKA youBot robotu kavrayıcısı, parmaklarının ortamdaki pozisyonlarını kontrol etmek için 6 farklı eylemi gerçekleştirmiştir. Hesaplama hızını artırmak için robotik kol önceden belirlenen yörünge sayesinde kavrama eylemini gerçekleştirebileceği alana getirilmektedir. Böylece, bütün bir robotik kolun her bir eklemi için ters kinematik hesaplama ihtiyacı ortadan kalkmaktadır. Kurulan ortam, robotik kolun kavrama sürecinde gerçek dünya senaryolarından ayrıldığı düşünülse de simülasyonda öyle olmadığı, Breyer ve arkadaşlarının yaptığı çalışmada da tüm kinematik zincir olmadan öğrenilen modelin gerçek dünya robotunda hala başarılı bir şekilde performans gösterebileceği belirtilmiştir [340]. Çalışmalarındaki öğrenme düzeni, kolun geri kalanı olmadan yalnızca bileğe kadar robot kavrayıcısı oluşturur. Böylece ters kinematik denklemi hesaplamaları gerekmez. Dolayısıyla bu tez çalışmasında da benzer yaklaşım izlenerek, robotik kolun tüm eklemleri için öğrenme düzeni kurmak yerine sadece robotik kolun uç işlevcisi olan kavrayıcı için öğrenme düzeni içeren bir pekiştirmeli öğrenme ortamı tanımlanmıştır.

Durum Uzayı

Ortam, gözlemlenen verileri işler ve her zaman adımından sonra ajana aktarır. Buradaki gözlem, ajanın Markov Karar Süreci'ndeki durumuna karşılık gelmektedir. Gözlem verileri Şekil 4.24'te görülen Kinect'in RGB kamerası vasıtasıyla alınmıştır. Kinect, bir RGB kameraya ve bir kızılötesi (IR) yayıcıya sahiptir. Sahnedeki her pikselin renkli bir görüntüsünü ve derinliğini yakalayabilirler. Bu veriler sahnenin görsel ve geometrik bilgilerini içerir. Görüntü işleme sistemlerinin çoğu, yalnızca görüntülerin renk kanallarına dayanmaktadır. Bu tez çalışmasında 150x100x3 boyutunda imgeler, durum uzay bilgisi için RGB kameradan alınmıştır. Kameradan alınan görüntülerin ajan tarafından durum bilgisi olarak işlenmesi için KSA tabanlı bir sinir ağı kullanılmıştır.



Şekil 4.24. Kinect Kamera yapısı

Eylem Uzayı

Ajanın tüm etkin eylemlerinin kümesine, eylem uzayı denir. Eylem uzayının bütünlük ve geçerlilik olarak iki özelliği olmalıdır. Bütünlük, ajanın beklenen hedefe ulaşma olasılığını garanti eder. Belirli bir eylem eksikse, aracının görevi tamamlamama olasılığı yüksektir. Örnek olarak; bir arabanın güvenliğini sağlamak için hızlanma, yavaşlama, dönme ve frenleme işlevlerine sahip olması gerekliliğidir. Geçerlilik, eylem uzayındaki eylemlerin yasal olmasını gerektirir. Mevcut yasalara göre araba sürmenin eylem uzayına asla uçma işlevi eklenmez.

Pekiştirmeli öğrenme, eylem uzayına göre ayırık eylem uzayı, sürekli eylem uzayı ve hibrit eylem uzayı olmak üzere üç ana kategoride incelenebilir. Ayırıklık ve süreklilik göreceli bir kavram olmakla birlikte ayırıklık sayılabilir, süreklilik sayılamaz. Ayırık eylemler için, eylemlerin sayısını açıkça belirtilebilir. Sürekli eylem uzayı probleminde, eylemlerin sayılamamasına rağmen belirli bir aralık tanımlanabilir. Pekiştirmeli öğrenmede, ayırık eylem uzaylarıyla ilgilenen algoritmalar çoğunlukla standart DQA, çift DQA ve düello DQA gibi DQÖ algoritmalarıdır. Bu tez çalışmasında eylem uzayı, düello DQA algoritmasının yapısına uygun şekilde belirli bir aralıkta tanımlanmıştır.

Ödül Fonksiyonun Yapısı

Ödül fonksiyonu, ödül ve ceza mekanizmasını kullanarak ajana neyin doğru neyin yanlış olduğunu söyleyen bir teşvik mekanizmasıdır. Ödül fonksiyonu, ortamın durumuna göre nümerik bir skor sağlar. Ödül fonksiyonu, ortamın algılanan her durumunun veya durum-eylem çiftinin tek bir sayıya eşlenmesidir. Pekiştirmeli öğrenmede ajanın amacı, toplam ödülleri en üst düzeye çıkarmaktır. Burada ödül fonksiyonu, nesneyi kavrama gibi ortamdaki tüm durumları ödüllendirirken, aynı zamanda eğitim sürecinin tamamlanmasından önce harcanan her zamanı cezalandırır. Ödül fonksiyonu tanımlanırken kavrayıcın her parmağı için hedef pozisyon aralığı belirlenmiştir. Kavrayıcı parmakları belirlenen pozisyon aralıklarına ulaşınca ödül değeri yüksek olmaktadır. Bu pozisyon aralıkları dışında kalınca negatif ödül değeri (ceza) almaktadır. Ödüller aşağıdaki gibi tanımlanmıştır.

Başarılı kavrama pozisyonu ödülü = +20 puan

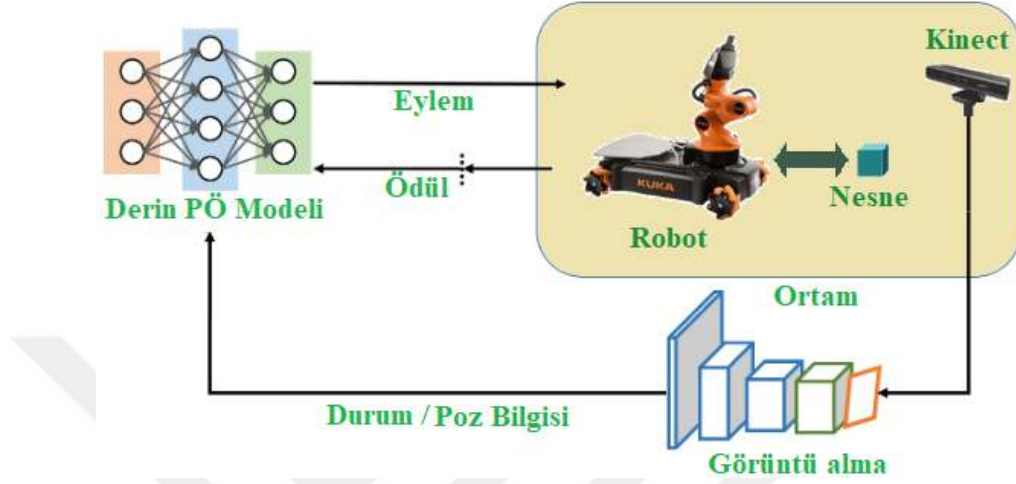
Başarısız kavrama ödülü pozisyonu = -20 puan

Harcanan her zaman adımı = -1 puan

Önerilen Öğrenme Yapısı

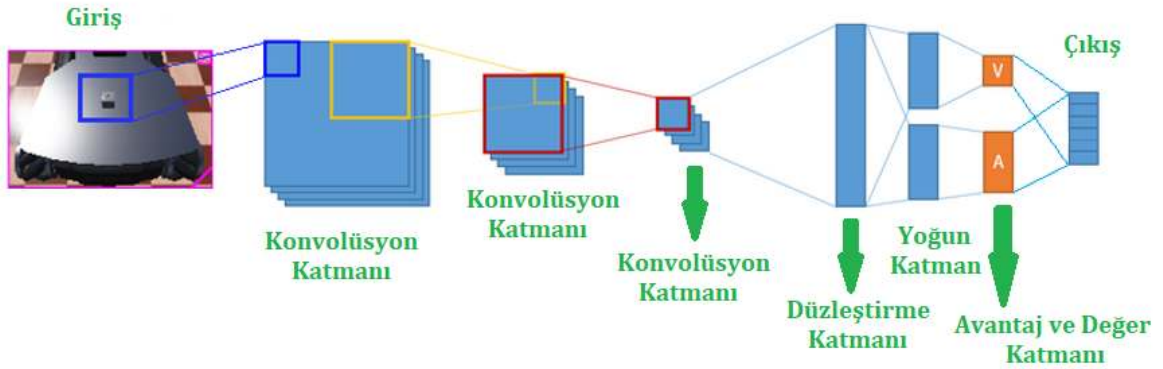
Görsel sensör olan kameralar, nesne ile teması algılamak ve kavrayıcılar ile kavranan nesneler arasındaki ilişkiyi izleyerek görsel geri bildirimle ortamla ilgili bilgi edinmek için önemli bir araçtır. Bu nedenle, kameraların kavrayıcılara entegre edilmesi, kavrama sürecini kontrol etmeyi kolaylaştırmaktadır. Bu tez çalışmasında Kinect'in VGA kamerasından alınan görüntüler düello DQN algoritmasına durum bilgisini aktarmak için kullanılmaktadır. Görüntüler KSA algoritmasıyla ajanın yapması gereken eylemleri tahmin etmektedir. Böylece KUKA youBot

robotunun uç işlevcisi kavrama hareketlerini gerçekleştirmektedir. Ödül fonksiyonu ile her yapılan eyleme karşılık bir ödül skoru belirlenmiştir. Kavrama süreci düello DQA algoritması kullanılarak bu şekilde devam etmektedir. Şekil 4.25’te önerilen pekiştirmeli öğrenmeye dayalı robotik kavrama yapısı görülmektedir.



Şekil 4.25. Önerilen pekiştirmeli öğrenmeye dayalı robotik kavrama yapısı

KSA algoritmasıyla ajanın gerçekleştireceği eylemler, Şekil 4.26’da Q-değerleri hesaplanarak tahmin edilmektedir. Q-değeri ($Q(s, a)$), s durumundaki bir ajanın eylemde bulunmasının ne kadar iyi olduğunu belirtir. Q değeri, o anki durumda olmanın değeri $V(s)$ ve diğer tüm olası eylemlerden o durumda o eylemi gerçekleştirmenin avantajı olan $A(s, a)$ toplamı olarak ayrıştırılabilir.



Şekil 4.26. Düello DKQA yapısında kullanılan adımlar

Deneyisel olarak elde edilen hiperparametrelerin sonuçları sunulmuştur. Kavrama performansı, kullanılan sinir ağının yığın boyutu büyüklüğünün etkisine bağlı olarak gözlemlenmiştir. Deneyisel olarak elde edilen hiperparametreler Tablo 4.11’de verilmiştir.

Tablo 4.11. Düello DQA Algoritması Hiperparametre Ayarları

Hiperparametre	Değer
Miniyığın	16-32
Replay Memory Boyutu	20000
Öğrenme Oranı	0.0001
İndirim Faktörü	0.95
ϵ	0.8
α	0.5
β	0.6

4.3.3. Uygulama Sonuçları

Bu bölümde, düello DQA algoritmasının uygulama sonuçları sunulmuştur. Senaryo #1 ve senaryo #2 olmak üzere iki ana başlık altında toplanan uygulama senaryolarının uygulanmasıyla elde edilen sonuçlar ayrı ayrı incelenmiştir. Senaryo #1’de KUKA youBot robotik kolunun sadece kavrayıcı parmaklarına ait eylem uzayı tanımlanırken senaryo #2’de tüm eklemler ve kavrayıcı parmaklarına ait bir eylem uzayı tanımlanmıştır. Kavrayıcı parmaklarının eylem planları, eylemleri birlikte yapılması istenirse kavrayıcı aynı anda açma ya da kapama hareketleri yapar, ayrı yapılması istenirse kavrayıcı parmakları birbirinden bağımsız olarak hareket eder. Her iki durumda da ödül fonksiyonuna bağlı olarak kavrayıcı parmaklarının en uygun pozisyona getirilmesi hedeflenmiştir.

Bu tez çalışmasındaki amaç, robotik kavrama görevlerinde gelişmiş DQA varyantlarından biri olan düello DQA algoritmasının performansını araştırmak olmuştur. Düello DQA algoritmasının hiperparametrelere ne kadar duyarlı olduğu ve verilen göreve ne kadar iyi adapte olabildiği irdelenmiştir. Normalde sürekli eylem alanına sahip olan robotik kavrama işleminde düello DQA algoritmasının nasıl performans gösterdiğine dair deneyimler elde edilmiştir. Hedef nesnenin yeri ve şekli, farklı senaryolar için değerlendirme süreci boyunca aynı tutulmuştur. Diğer bir deyişle, önerilen model aynı koşullarda test edilmiştir. Kavrayıcı parmaklarını önceden tanımlanmış pozisyonlara getirebildiğinde, yani nesne kavrayabilme pozisyonuna erişildiğinde bir bölüm başarılı olarak kabul edilmiştir.

Uygulama çalışmalarında basitleştirilmiş ortamın sonuçları gösterilmiştir ve hiperparametre değişikliklerine dayalı farklılıklar gözlemlenmiştir. Kullanılan sinir ağının katman sayısı ve her katmandaki nöron sayısı, eğitim adım sayısı, öğrenme oranı, yığın boyutu ve hedef ağın güncellenmesi gibi durumların etkileri irdelenmiştir.

Senaryo 1 Sonuçları

Senaryo #1’de KUKA youBot robotik kolunun kavrayıcı parmaklarına ait eylem uzayı tanımlanmıştır. Durum uzayı bilgisi için 150x100x3 boyutunda imgeler RGB kameradan alınarak elde edilmiştir. Tasarlanan düello DQA algoritmasıyla ajanın hareket politikası belirlenmiştir. Ödül fonksiyonu her iki senaryo için aynı tutulmakla birlikte eklemlerdeki hareketlerde göz önünde bulundurularak ödül fonksiyonu daha geniş bir şekilde düzenlenmiştir. Kavrayıcı ile hedef nesne arasındaki uzaklığa bağlı olarak her bir eyleme ait ödüller tanımlanmıştır. Durum bilgisi alınırken imgelerdeki özniteliklerin elde edilmesi için KSA modeli oluşturulmuştur.

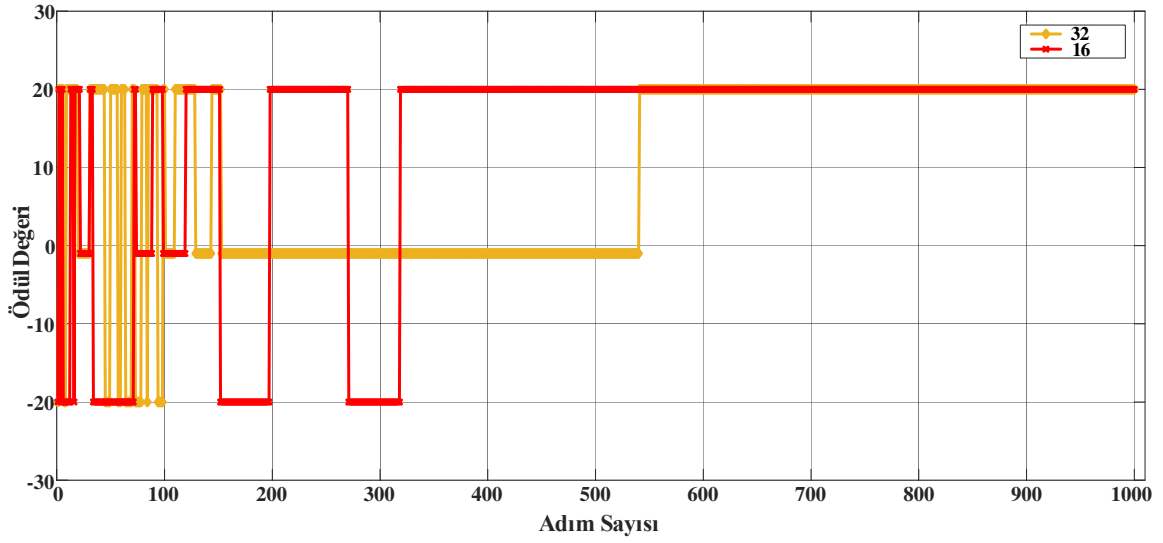
Senaryo #1 ve Senaryo #2’de, eylem seçimi epsilon greedy keşif stratejisine dayalı olarak Şekil 4.27’deki kod parçacığı ile yapılmıştır.

```
44 def policy_action(self, s):
45     """ An epsilon-greedy policy to pick next action
46     """
47     if random() <= self.epsilon:
48         return randrange(self.action_dim)
49     else:
50         return np.argmax(self.agent.predict(s)[0])
51
75 # Decay epsilon
76 self.epsilon *= self.epsilon_decay
77
```

Şekil 4.27. Eylem seçimi kod parçacığı

Epsilon greedy keşif stratejisi, bir keşif sömürü ödünleşimi yaklaşımıdır. Düello DQA algoritması (ajan) başlangıçta yapmış olduğu her bir eylemi rastgele seçmektedir. Bu da ajanın içinde bulunduğu ortamı keşfetmesine imkân sağlamaktadır. Ajan eylem gerçekleştirdikçe epsilon değeri düşmekte ve ajan öğrendiği mevcut iyi eylemleri denemeye başlamaktadır.

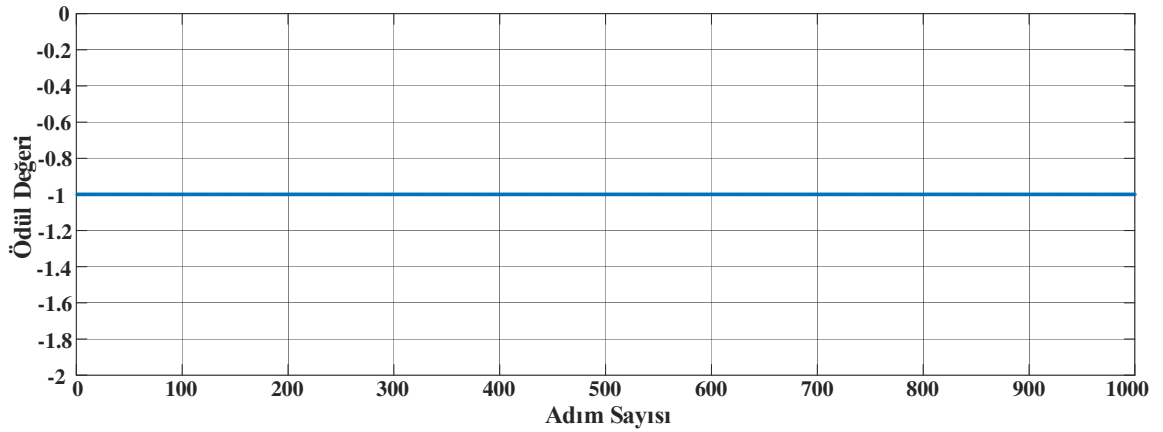
Senaryo #1’de epsilon değeri başlangıçta eylemler rasgele seçilebilmesi için yüksek seçilir. Şekilde elde edilen sonuçta epsilon değeri 0.8 ve epsilon azalma değeri 0.99 olarak alınmıştır. Böylece ajan, eğitim aşamasının sonuna doğru daha az keşif daha fazla sömürü yaklaşımını izleyerek kavrama eylemini öğrenmiştir. Şekil 4.28’de yığın boyutu 16 ve 32 alınarak sonuçlara bakıldığında yığın boyutu 16 olduğunda doğru eylem seçimi sürekli hale geçtiği ve ajanın sömürü yaklaşımını izlediği görülmektedir.



Şekil 4.28. Anlık ödül değeri değişimi

Senaryo 2 Sonuçları

Senaryo #2’de KUKA youBot robotik kolunun tüm eklemlerine ve kavrayıcı parmaklarına ait eylem uzayı tanımlanmıştır. Durum uzayı bilgisi için 150x100x3 boyutunda imgeler RGB kameradan alınarak elde edilmiştir. Ajanın hareket politikası, tasarlanan düello DQA algoritmasıyla belirlenmiştir. Ödül fonksiyonu Senaryo #1 ile aynı tutulmakla birlikte eklemlerdeki hareketlerde göz önünde bulundurularak daha geniş bir şekilde düzenlenmiştir. Kavrayıcı ile hedef nesne arasındaki uzaklığa bağlı olarak her bir eyleme ait ödüller tanımlanmıştır. Durum bilgisi alınırken imgelerdeki özniteliklerin elde edilmesi için KSA modeli oluşturulmuştur.



Şekil 4.29. Anlık ödül değeri değişimi

Şekil 4.29’da elde edilen sonuca bakıldığında aynı öğrenme yaklaşımıyla herhangi bir ödül değerinin üretilmediği görülmüştür. Bu sonuçla eylem sayısının artması durumunda öğrenme olayının gerçekleştirilemediği sonucuna varılmıştır.

5. SONUÇLAR

Bu tez çalışması, ulusal ve uluslararası alanda gittikçe popüler ve önemli bir teknoloji haline gelmekte olan nesne kavrama ve yapay zekâ konusunda model geliştirme ve uygulama çalışmalarını kapsamaktadır. Öncelikle robotik kavrama probleminin modellenmesi aşamalarına geçmeden, insan kavraması üzerine bir derin öğrenme çalışması gerçekleştirilmiştir. Bunun için 6 sağlıklı bireye ait veriler kullanılarak, derin öğrenme mimarisi ile otomatik sınıflandırma problemi ele alınmıştır. Tablo 4.8’de verilen referans çalışmalarla karşılaştırıldığında elde edilen sonuçlar doğrultusunda geliştirilen 1B-KSA modeli, yEMG verileri üzerinde sınıflandırma işlemini %94.94 gibi yüksek bir başarımla gerçekleştirdiği gözlemlenmiştir.

Bu tezin ana konusu olan DPÖ yöntemlerinin robotik kavrama probleminde kullanımı için bir ön çalışma yapılmıştır. Bunun için önce, oyun oynayabilen DKQÖ tabanlı bir yapay zekâ ajanının/modelinin eğitilmesi süreci ele alınmıştır. Daha sonra simülasyon ortamında düello DQA algoritmasıyla robotik kavrama senaryosu oluşturulmuştur. Elde edilen sonuçlardan, bir robotik kavrama uygulamasının gerçekleştirilmesinde ayrı bir yapıya dayalı olan düello DQA algoritmasının tatmin edici performansla sürekli eylem uzaylarına başarıyla ölçeklenebileceği gösterilmiştir. Ancak, düello DQA’nın artan eylem uzayı boyutlarıyla başa çıkamadığı gözlemlenmiştir.

Bu tez çalışmasında, DPÖ’ye dayalı bir robotik kavrama görevinin simülasyonu için temel bir yapı sunulmuştur. Ajan tarafından ortamda gerçekleştirilen sürekli eylemler sorunuyla başa çıkmak için yeni bir ödül fonksiyonu tasarlanmıştır. Ayrıca önerilen düello DQA algoritması için tatmin edici hiper parametreler elde edilmiştir. Bu çalışmadan elde edilen önemli bir katkı, optimal bir yığın boyutu büyüklüğü elde etme rolüdür. Yığın boyutu göz önüne alındığında, 32’ye kıyasla 16 mini yığın boyutunun daha iyi sonuçlar verdiği gözlemlenmiştir.

Bu tez çalışması ile birçok uygulamada karşılaşılan nesne kavrama becerileri geliştirilmiştir ve bu doğrultuda KSA ve DPÖ temelli farklı algoritmalar sunulmuştur. Tezde endüstrideki mevcut problemler belirlendikten sonra nesne kavramaya yönelik sınıflandırma ve kontrol becerileri geliştirilmiştir. Özellikle endüstriyel alanın bazı süreçleri için kullanıcıların problemlerine çözüm önerileri sunulmuştur.

ÖNERİLER VE GELECEK ÇALIŞMALAR

Yapılan bu tez çalışmasından elde edilen tecrübelerle, ileri ve diğer güncel derin pekiştirmeli öğrenme yöntemler ile robotik kavrama yetenekleri geliştirilebilir. Bu tez çalışmasında iki parmaklı/paralel kavrayıcı ile kavrama eylemi gerçekleştirilmiştir. Ayrıca, 6 sağlıklı bireye ait kavrama tipleri otomatik olarak sınıflandırılmıştır. Gelecek çalışmalarda tez kapsamında yapılan çalışmalar ışığında simülasyon ortamında beş parmaklı/antromorfik el ile kavrama senaryoları geliştirilecektir. Ayrıca gerçek robot üzerinde bu yöntemlerin performansları incelenebilir. Simülasyon modellerini donanıma aktarmak, sensör gürültüsü veya robot kalibrasyonu gibi yeni zorlukları beraberinde getirebilir. Bu tez çalışmasında, sadece katı bir nesnenin kavranması planlanmıştır. Gelecekte nesnelerin sayısını artırmayı ve deforme olabilen nesneleri kavramaya yönelik yaklaşımlar ele alınabilir.

Bu tez çalışmasında, pekiştirmeli öğrenme modeli sadece simülasyon ortamında test edilmiştir. Simülasyon modellerini donanıma aktarmak, sensör gürültüsü veya robot kalibrasyonu gibi bazı zorluklara neden olabilir. Donanımsal zorluklara rağmen gelecekte kavrama politikasının simülasyondan donanıma aktarılmasının sağlanması arzulanmaktadır. Simülasyonda oluşturulan senaryoda düello DQA algoritması ile tatmin edici sonuçlar elde edilmiştir. Ancak robotik kol için tam eylem uzayı kullanıldığında kötü performans gösterdiği görülmüştür. Bu başarısızlığın nedeni, yetersiz keşif kabiliyetidir. Gelecek çalışmalarda bu başarısızlığa çözüm olabilecek politikaya dayalı derin PÖ yöntemleri incelenecektir. Politikaya dayalı DPÖ yöntemlerinin DQA ve türlerine göre avantajı, daha iyi keşif stratejileri sağlayabilmesidir.

KAYNAKLAR

- [1] Y. Bengio, Y. Lecun, G. Hinton, Deep learning for AI, *Commun. ACM.* 64 (2021) 58–65. <https://doi.org/10.1145/3448250>.
- [2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature.* 521 (2015) 436–444. <https://doi.org/10.1038/nature14539>.
- [3] K. Kleeberger, R. Bormann, W. Kraus, M.F. Huber, A Survey on Learning-Based Robotic Grasping, *Curr. Robot. Reports.* 1 (2020) 239–249. <https://doi.org/10.1007/s43154-020-00021-6>.
- [4] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F.R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. Chavan Dafle, R. Holladay, I. Morona, P.Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, A. Rodriguez, Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching, *Int. J. Rob. Res.* 41 (2022) 690–705. <https://doi.org/10.1177/0278364919868017>.
- [5] S. Kumra, C. Kanan, Robotic grasp detection using deep convolutional neural networks, in: 2017 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2017: pp. 769–776. <https://doi.org/10.1109/IROS.2017.8202237>.
- [6] M. El-Shamouty, K. Kleeberger, A. Lämmle, M. Huber, Simulation-driven machine learning for robotics and automation, *Tm - Tech. Mess.* 86 (2019) 673–684. <https://doi.org/10.1515/teme-2019-0072>.
- [7] J. Bohg, A. Morales, T. Asfour, D. Kragic, Data-Driven Grasp Synthesis—A Survey, *IEEE Trans. Robot.* 30 (2014) 289–309. <https://doi.org/10.1109/TRO.2013.2289018>.
- [8] A. Sahbani, S. El-Khoury, P. Bidaud, An overview of 3D object grasp synthesis algorithms, *Rob. Auton. Syst.* 60 (2012) 326–336. <https://doi.org/10.1016/j.robot.2011.07.016>.
- [9] R. Bormann, B.F. de Brito, J. Lindermayr, M. Omainiska, M. Patel, Towards Automated Order Picking Robots for Warehouses and Retail, in: 2019: pp. 185–198. https://doi.org/10.1007/978-3-030-34995-0_18.
- [10] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J.A. Ojea, K. Goldberg, Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, (2017). <http://arxiv.org/abs/1703.09312>.
- [11] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, K. Goldberg, Dex-Net 3.0: Computing Robust Robot Vacuum Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning, (2017). <http://arxiv.org/abs/1709.06670>.
- [12] J. Redmon, A. Angelova, Real-Time Grasp Detection Using Convolutional Neural Networks, (2014). <http://arxiv.org/abs/1412.3128>.
- [13] D. Morrison, P. Corke, J. Leitner, Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach, (2018). <http://arxiv.org/abs/1804.05172>.
- [14] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, K. Bousmalis, Sim-To-Real via Sim-To-Sim: Data-Efficient Robotic Grasping via Randomized-To-Canonical Adaptation Networks, in: 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., IEEE, 2019: pp. 12619–12629. <https://doi.org/10.1109/CVPR.2019.01291>.
- [15] O.K. Bruno Siciliano, Springer Handbook of Robotics, 2008.
- [16] L. Pinto, A. Gupta, Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours, (2015). <http://arxiv.org/abs/1509.06825>.
- [17] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, D. Quillen, Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, *Int. J. Rob. Res.* 37 (2018) 421–436. <https://doi.org/10.1177/0278364917710318>.
- [18] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, S. Levine, QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation, (2018). <http://arxiv.org/abs/1806.10293>.
- [19] E. Rohmer, S.P.N. Singh, M. Freese, V-REP: A versatile and scalable robot simulation framework, in: 2013 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2013: pp. 1321–1326. <https://doi.org/10.1109/IROS.2013.6696520>.

- [20] S. James, M. Freese, A.J. Davison, PyRep: Bringing V-REP to Deep Robot Learning, (2019). <http://arxiv.org/abs/1906.11176>.
- [21] E. Coumans and Y. Bai, Pybullet, a python module for physics simulation for games, robotics and machine learning, (2019). <http://pybullet.org>.
- [22] E. Todorov, T. Erez, Y. Tassa, MuJoCo: A physics engine for model-based control, in: 2012 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2012: pp. 5026–5033. <https://doi.org/10.1109/IROS.2012.6386109>.
- [23] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IEEE Cat. No.04CH37566), IEEE, n.d.: pp. 2149–2154. <https://doi.org/10.1109/IROS.2004.1389727>.
- [24] O. Michel, Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation, Int. J. Adv. Robot. Syst. 1 (2004) 5. <https://doi.org/10.5772/5618>.
- [25] Peter Corke, Robotics, Vision and Control Fundamental Algorithms in MATLAB, 2nd ed., 2017.
- [26] Definition of robot, (2022). <https://www.merriam-webster.com/dictionary/robot>.
- [27] C.G.D. Considine D.M., Standard Handbook of Industrial Automation, Robot Technology Fundamentals, Springer, Boston, MA, 1986. https://doi.org/10.1007/978-1-4613-1963-4_17.
- [28] Isaac Asimov, I, Robot, Book Club Edition, 1950.
- [29] Susan Leigh Anderson, Asimov’s “Three Laws of Robotics” and Machine Metaethics, 2016.
- [30] Y. hao Li, Q. jiang Lei, C. Cheng, G. Zhang, W. Wang, Z. Xu, A review: machine learning on robotic grasping, in: D.P. Nikolaev, P. Radeva, A. Verikas, J. Zhou (Eds.), Elev. Int. Conf. Mach. Vis. (ICMV 2018), SPIE, 2019: p. 54. <https://doi.org/10.1117/12.2522945>.
- [31] William Rowan Hamilton, On quaternions; or on a new system of imaginaries in algebra, (2009) 10–13.
- [32] R. Featherstone, The Calculation of Robot Dynamics Using Articulated-Body Inertias, Int. J. Rob. Res. 2 (1983) 13–30.
- [33] D.T. Greenwood, Classical Dynamics, 1997.
- [34] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, The MIT Press, 2010.
- [35] J.H. Reif, Complexity of the mover’s problem and generalizations, in: 20th Annu. Symp. Found. Comput. Sci. (Sfcs 1979), IEEE, 1979: pp. 421–427. <https://doi.org/10.1109/SFCS.1979.10>.
- [36] G.B. Carlos Canudas Wit, Bruno Siciliano, Theory of Robot Control, 1996.
- [37] D.W. T. Fazio, Donald S. Seltzer, The instrumented remote centre compliance, Ind. Robot. Int. J. (1984).
- [38] G. Zeng, And, A. Hemami, An overview of robot force control, Robotica. 15 (1997) 473–482.
- [39] J.R. I. Jacobson, G. Booch, Unified Software Development Process, Addison-Wesley Professional, 1999.
- [40] P.P.-S. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, The Description Logic Handbook, Cambridge University Press, 2007. <https://doi.org/10.1017/CBO9780511711787>.
- [41] P.N. Stuart Russell, Artificial Intelligence: A Modern Approach, 3rd ed., Pearson, 2009.
- [42] P.T. Malik Ghallab, Dana Nau, Automated Planning: Theory & Practice (The Morgan Kaufmann Series in Artificial Intelligence), Morgan Kaufmann, 2004.
- [43] Jussi Rintanen and Jörg Hoffmann, An overview of recent algorithms for AI planning, Künstliche Intelligenz. 2 (2001) 5–11.
- [44] I.H. Sarker, Machine Learning: Algorithms, Real-World Applications and Research Directions, SN Comput. Sci. 2 (2021) 160. <https://doi.org/10.1007/s42979-021-00592-x>.
- [45] S.D. Nada Lavrač, Inductive Logic Programming: Techniques and Applications, 1994.
- [46] P.P. Shinde, S. Shah, A Review of Machine Learning and Deep Learning Applications, in: 2018 Fourth Int. Conf. Comput. Commun. Control Autom., IEEE, 2018: pp. 1–6. <https://doi.org/10.1109/ICCUBEA.2018.8697857>.
- [47] K.J. Danjuma, Performance Evaluation of Machine Learning Algorithms in Post-operative Life Expectancy in the Lung Cancer Patients, (2015). <http://arxiv.org/abs/1504.04646>.
- [48] F. Zahedi, An Introduction to Neural Networks and a Comparison with Artificial Intelligence and

- Expert Systems, Interfaces (Providence). 21 (1991) 25–38. <https://doi.org/10.1287/inte.21.2.25>.
- [49] A. Sutton, R.; Barto, Reinforcement Learning, 2nd ed., MIT Press, 1998.
- [50] C.J. Watkins, Models of Delayed Reinforcement Learning, Cambridge, 1989.
- [51] Y. Oh, Y. Kim, K. Na, B.D. Youn, A deep transferable motion-adaptive fault detection method for industrial robots using a residual–convolutional neural network, *ISA Trans.* (2021). <https://doi.org/10.1016/j.isatra.2021.11.019>.
- [52] S.Y. Nof, Handbook of Industrial Robotics, 2nd ed., Wiley, 1999.
- [53] A. Dzedzickis, J. Subačiūtė-Žemaitienė, E. Šutinys, U. Samukaitė-Bubnienė, V. Bučinskas, Advanced Applications of Industrial Robotics: New Trends and Possibilities, *Appl. Sci.* 12 (2021) 135. <https://doi.org/10.3390/app12010135>.
- [54] M. Bertolini, D. Mezzogori, M. Neroni, F. Zammori, Machine Learning for industrial applications: A comprehensive literature review, *Expert Syst. Appl.* 175 (2021) 114820. <https://doi.org/10.1016/j.eswa.2021.114820>.
- [55] H. Zhang, Z. Chen, C. Zhang, J. Xi, X. Le, Weld Defect Detection Based on Deep Learning Method, in: 2019 IEEE 15th Int. Conf. Autom. Sci. Eng., IEEE, 2019: pp. 1574–1579. <https://doi.org/10.1109/COASE.2019.8842998>.
- [56] R. Espinosa, H. Ponce, S. Gutiérrez, Click-event sound detection in automotive industry using machine/deep learning, *Appl. Soft Comput.* 108 (2021) 107465. <https://doi.org/10.1016/j.asoc.2021.107465>.
- [57] C.A. Woolsey, Review of Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles, *J. Guid. Control. Dyn.* 28 (2005) 574–575. <https://doi.org/10.2514/1.17190>.
- [58] S. Chutia, N.M. Kakoty, D. Deka, A Review of Underwater Robotics, Navigation, Sensing Techniques and Applications, in: *Proc. Adv. Robot. - AIR '17*, ACM Press, New York, New York, USA, 2017: pp. 1–6. <https://doi.org/10.1145/3132446.3134872>.
- [59] J. Neira, C. Sequeiros, R. Huamani, E. Machaca, P. Fonseca, W. Nina, Review on Unmanned Underwater Robotics, Structure Designs, Materials, Sensors, Actuators, and Navigation Control, *J. Robot.* 2021 (2021) 1–26. <https://doi.org/10.1155/2021/5542920>.
- [60] Y. Cong, C. Gu, T. Zhang, Y. Gao, Underwater robot sensing technology: A survey, *Fundam. Res.* 1 (2021) 337–345. <https://doi.org/10.1016/j.fmre.2021.03.002>.
- [61] M. Manjunatha, A.A. Selvakumar, V.P. Godeswar, R. Manimaran, A Low Cost Underwater Robot with Grippers for Visual Inspection of External Pipeline Surface, *Procedia Comput. Sci.* 133 (2018) 108–115. <https://doi.org/10.1016/j.procs.2018.07.014>.
- [62] A. Sahoo, S.K. Dwivedy, P.S. Robi, Advancements in the field of autonomous underwater vehicle, *Ocean Eng.* 181 (2019) 145–160. <https://doi.org/10.1016/j.oceaneng.2019.04.011>.
- [63] R.C. Michelson, International aerial robotics competition - The world's smallest intelligent flying machines, in: 13th Bristol Int. RPV/UAV Syst. Conf. Proc., 1998: pp. 31.1–31.9.
- [64] S.-J. Chung, A.A. Paranjape, P. Dames, S. Shen, V. Kumar, A Survey on Aerial Swarm Robotics, *IEEE Trans. Robot.* 34 (2018) 837–855. <https://doi.org/10.1109/TRO.2018.2857475>.
- [65] X. Wei-hong, C. Li-jia, Z. Chun-lai, Review of Aerial Manipulator and its Control, *Int. J. Robot. Control Syst.* 1 (2021) 308–325. <https://doi.org/10.31763/ijrcs.v1i3.363>.
- [66] R. Maity, R. Mishra, P. Kumar Pattnaik, Flying robot path planning techniques and its trends, *Mater. Today Proc.* (2021). <https://doi.org/10.1016/j.matpr.2021.06.174>.
- [67] C.R.C. D. L. Akin, M. L. Minsky, E. D. Thiel, Space applications of automation, robotics and machine intelligence systems, 1983.
- [68] M.F. Stieber, C.P. Trudel, D.G. Hunter, Robotic systems for the International Space Station, in: *Proc. Int. Conf. Robot. Autom.*, IEEE, n.d.: pp. 3068–3073. <https://doi.org/10.1109/ROBOT.1997.606754>.
- [69] B. Zhang, Y. Xie, J. Zhou, K. Wang, Z. Zhang, State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review, *Comput. Electron. Agric.* 177 (2020) 105694. <https://doi.org/10.1016/j.compag.2020.105694>.
- [70] J.J. Roldán, J. del Cerro, D. Garzón-Ramos, P. Garcia-Aunon, M. Garzón, J. de León, A. Barrientos, Robots in Agriculture: State of Art and Practical Experiences, in: *Serv. Robot.*, InTech, 2018. <https://doi.org/10.5772/intechopen.69874>.

- [71] E. Ginzberg, The Mechanization of Work, *Sci. Am.* 247 (1982) 66–75.
- [72] V. Frangez, E. Lloret-Fritschi, N. Taha, F. Gramazio, M. Kohler, A. Wieser, Depth-camera-based rebar detection and digital reconstruction for robotic concrete spraying, *Constr. Robot.* 5 (2021) 191–202. <https://doi.org/10.1007/s41693-021-00063-y>.
- [73] M. Gharbia, A. Chang-Richards, Y. Lu, R.Y. Zhong, H. Li, Robotic technologies for on-site building construction: A systematic review, *J. Build. Eng.* 32 (2020) 101584. <https://doi.org/10.1016/j.jobte.2020.101584>.
- [74] N.S. Plotnikov, E.U. Kolokoltseva, Y. V Volkova, Technical Review of Robotic Complexes for Underground Mining, *IOP Conf. Ser. Earth Environ. Sci.* 459 (2020) 042025. <https://doi.org/10.1088/1755-1315/459/4/042025>.
- [75] A.H. Reddy, B. Kalyan, C.S.N. Murthy, Mine Rescue Robot System – A Review, *Procedia Earth Planet. Sci.* 11 (2015) 457–462. <https://doi.org/10.1016/j.proeps.2015.06.045>.
- [76] D. Paredes, D. Fleming-Muñoz, Automation and robotics in mining: Jobs, income and inequality implications, *Extr. Ind. Soc.* 8 (2021) 189–193. <https://doi.org/10.1016/j.exis.2021.01.004>.
- [77] G. Singh, V.K. Banga, Robots and its types for industrial applications, *Mater. Today Proc.* 60 (2022) 1779–1786. <https://doi.org/10.1016/j.matpr.2021.12.426>.
- [78] B. Davies, A review of robotics in surgery, *Proc. Inst. Mech. Eng. Part H J. Eng. Med.* 214 (2000) 129–140. <https://doi.org/10.1243/0954411001535309>.
- [79] K. Cleary, C. Nguyen, State of the art in surgical robotics: Clinical applications and technology challenges, *Comput. Aided Surg.* 6 (2001) 312–328. <https://doi.org/10.1002/igs.10019>.
- [80] J. Speich, J. Rosen, Medical Robotics, in: *Encycl. Biomater. Biomed. Eng.* Second Ed. - Four Vol. Set, CRC Press, 2008: pp. 1804–1815. <https://doi.org/10.1201/b18990-174>.
- [81] G. Dogangil, B.L. Davies, F. Rodriguez y Baena, A review of medical robotics for minimally invasive soft tissue surgery, *Proc. Inst. Mech. Eng. Part H J. Eng. Med.* 224 (2010) 653–679. <https://doi.org/10.1243/09544119JEIM591>.
- [82] R.H. Taylor, B.D. Mittelstadt, H.A. Paul, W. Hanson, P. Kazanzides, J.F. Zuhars, B. Williamson, B.L. Musits, E. Glassman, W.L. Bargar, An image-directed robotic system for precise orthopaedic surgery, *IEEE Trans. Robot. Autom.* 10 (1994) 261–275. <https://doi.org/10.1109/70.294202>.
- [83] P. Kazanzides, B.D. Mittelstadt, B.L. Musits, W.L. Bargar, J.F. Zuhars, B. Williamson, P.W. Cain, E.J. Carbone, An integrated system for cementless hip replacement, *IEEE Eng. Med. Biol. Mag.* 14 (1995) 307–313. <https://doi.org/10.1109/51.391772>.
- [84] K. Corker, J.H. Lyman, S. Sheredos, A preliminary evaluation of remote medical manipulators., *Bull. Prosthet. Res.* 16 (1979) 107–34. <http://www.ncbi.nlm.nih.gov/pubmed/519083>.
- [85] R. Gassert, V. Dietz, Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective, *J. Neuroeng. Rehabil.* 15 (2018) 46. <https://doi.org/10.1186/s12984-018-0383-x>.
- [86] E.D. Oña, R. Cano-de la Cuerda, P. Sánchez-Herrera, C. Balaguer, A. Jardón, A Review of Robotics in Neurorehabilitation: Towards an Automated Process for Upper Limb, *J. Healthc. Eng.* 2018 (2018) 1–19. <https://doi.org/10.1155/2018/9758939>.
- [87] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H.R. Nambiappan, K.K. Chaitanya, A.R. Babu, J. Mathew, F. Makedon, A Survey of Robots in Healthcare, *Technologies.* 9 (2021) 8. <https://doi.org/10.3390/technologies9010008>.
- [88] A. Abou Allaban, M. Wang, T. Padir, A Systematic Review of Robotics Research in Support of In-Home Care for Older Adults, *Information.* 11 (2020) 75. <https://doi.org/10.3390/info11020075>.
- [89] C. Caudwell, C. Lacey, What do home robots want? The ambivalent power of cuteness in robotic relationships, *Converg. Int. J. Res. into New Media Technol.* 26 (2020) 956–968. <https://doi.org/10.1177/1354856519837792>.
- [90] A. Langer, R. Feingold-Polak, O. Mueller, P. Kellmeyer, S. Levy-Tzedek, Trust in socially assistive robots: Considerations for use in rehabilitation, *Neurosci. Biobehav. Rev.* 104 (2019) 231–239. <https://doi.org/10.1016/j.neubiorev.2019.07.014>.
- [91] A. Takanishi, Historical Perspective of Humanoid Robot Research in Asia, in: *Humanoid Robot. A Ref.*, Springer Netherlands, Dordrecht, 2019: pp. 35–52. https://doi.org/10.1007/978-94-007-6046-2_145.
- [92] M. Vukobratović, B. Borovac, Zero-Moment Point — Thirty Five Years of Its Life, *Int. J. Humanoid*

- Robot. 01 (2004) 157–173. <https://doi.org/10.1142/S0219843604000083>.
- [93] and R.L. C. Laschi, P. Dario, M. Carrozza, E. Guglielmelli, G. Teti, D. Taddeucci, F. Leoni, B. Massa, M. Zecca, Grasping and manipulation in humanoid robotics, in: IEEE Int. Conf. Humanoid Robot., 2003.
- [94] T. McGeer, Passive Dynamic Walking, Int. J. Rob. Res. 9 (1990) 62–82. <https://doi.org/10.1177/027836499000900206>.
- [95] T. Zielinska, Development of Walking Machines; Historical Perspective, in: Int. Symp. Hist. Mach. Mech., Springer Netherlands, Dordrecht, n.d.: pp. 357–370. https://doi.org/10.1007/1-4020-2204-2_28.
- [96] P. Biswal, P.K. Mohanty, Development of quadruped walking robots: A review, Ain Shams Eng. J. 12 (2021) 2017–2031. <https://doi.org/10.1016/j.asej.2020.11.005>.
- [97] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, The development of Honda humanoid robot, in: Proceedings. 1998 IEEE Int. Conf. Robot. Autom. (Cat. No.98CH36146), IEEE, n.d.: pp. 1321–1326. <https://doi.org/10.1109/ROBOT.1998.677288>.
- [98] S. Shigemi, ASIMO and Humanoid Robot Research at Honda, in: Humanoid Robot. A Ref., Springer Netherlands, Dordrecht, 2019: pp. 55–90. https://doi.org/10.1007/978-94-007-6046-2_9.
- [99] K. Nagasaka, Sony QRIO, in: Humanoid Robot. A Ref., Springer Netherlands, Dordrecht, 2019: pp. 187–200. https://doi.org/10.1007/978-94-007-6046-2_16.
- [100] S. Kajita, HRP Robots (e.g. HRP-4), in: Humanoid Robot. A Ref., Springer Netherlands, Dordrecht, 2017: pp. 1–16. https://doi.org/10.1007/978-94-007-7194-9_11-1.
- [101] S. Shamsuddin, L.I. Ismail, H. Yussof, N. Ismarrubie Zahari, S. Bahari, H. Hashim, A. Jaffar, Humanoid robot NAO: Review of control and motion exploration, in: 2011 IEEE Int. Conf. Control Syst. Comput. Eng., IEEE, 2011: pp. 511–516. <https://doi.org/10.1109/ICCSCE.2011.6190579>.
- [102] J.K. Salisbury, B. Roth, Kinematic and Force Analysis of Articulated Mechanical Hands, J. Mech. Transm. Autom. Des. 105 (1983) 35–41. <https://doi.org/10.1115/1.3267342>.
- [103] H. Duan, P. Wang, Y. Huang, G. Xu, W. Wei, X. Shen, Robotics Dexterous Grasping: The Methods Based on Point Cloud and Deep Learning, Front. Neurobot. 15 (2021). <https://doi.org/10.3389/fnbot.2021.658280>.
- [104] G. Du, K. Wang, S. Lian, K. Zhao, Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review, Artif. Intell. Rev. 54 (2021) 1677–1734. <https://doi.org/10.1007/s10462-020-09888-5>.
- [105] S. Caldera, A. Rassau, D. Chai, Review of Deep Learning Methods in Robotic Grasp Detection, Multimodal Technol. Interact. 2 (2018) 57. <https://doi.org/10.3390/mti2030057>.
- [106] Yun Jiang, S. Moseson, A. Saxena, Efficient grasping from RGBD images: Learning using a new rectangle representation, in: 2011 IEEE Int. Conf. Robot. Autom., IEEE, 2011: pp. 3304–3311. <https://doi.org/10.1109/ICRA.2011.5980145>.
- [107] I. Lenz, H. Lee, A. Saxena, Deep learning for detecting robotic grasps, Int. J. Rob. Res. 34 (2015) 705–724. <https://doi.org/10.1177/0278364914549607>.
- [108] A. Saxena, J. Driemeyer, A.Y. Ng, Robotic Grasping of Novel Objects using Vision, Int. J. Rob. Res. 27 (2008) 157–173. <https://doi.org/10.1177/0278364907087172>.
- [109] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, S. Schaal, Learning of grasp selection based on shape-templates, Auton. Robots. 36 (2014) 51–65. <https://doi.org/10.1007/s10514-013-9366-8>.
- [110] D. Guo, F. Sun, T. Kong, H. Liu, Deep vision networks for real-time robotic grasp detection, Int. J. Adv. Robot. Syst. 14 (2017) 172988141668270. <https://doi.org/10.1177/1729881416682706>.
- [111] K. Tai, A.-R. El-Sayed, M. Shahriari, M. Biglarbegian, S. Mahmud, State of the Art Robotic Grippers and Applications, Robotics. 5 (2016) 11. <https://doi.org/10.3390/robotics5020011>.
- [112] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M.R. Zakin, H. Lipson, H.M. Jaeger, Universal robotic gripper based on the jamming of granular material, Proc. Natl. Acad. Sci. 107 (2010) 18809–18814. <https://doi.org/10.1073/pnas.1003250107>.
- [113] Patakota Venkata Prasad Reddy and V V N Satya Suresh, A Review on Importance of Universal Gripper in Industrial Robot Applications, Int. J. Mech. Eng. Robot. Res. 2 (2013) 255–264. <https://doi.org/10.18178/ijmerr>.

- [114] T.K. Lien, Gripper technologies for food industry robots, in: *Robot. Autom. Food Ind.*, Elsevier, 2013: pp. 143–170. <https://doi.org/10.1533/9780857095763.1.143>.
- [115] H. Monkman, G.J., Hesse, S., Steinmann, R., Schunk, *Robot Grippers*, Wiley-VCH, 2006.
- [116] X. Li, T. Kagawa, Theoretical and Experimental Study of Factors Affecting the Suction Force of a Bernoulli Gripper, *J. Eng. Mech.* 140 (2014). [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0000774](https://doi.org/10.1061/(ASCE)EM.1943-7889.0000774).
- [117] H.S. Andreas Wolf, Ralf Steinmann, *Grippers in Motion*, Springer, 2006.
- [118] B.P. Samir Raval, A Review on Grasping Principle and Robotic Grippers, *Int. J. Eng. Dev. Res.* 4 (2016) 483–490.
- [119] L. Birglen, T. Schlicht, A statistical review of industrial robotic grippers, *Robot. Comput. Integr. Manuf.* 49 (2018) 88–97. <https://doi.org/10.1016/j.rcim.2017.05.007>.
- [120] P.O. Hugo, Industrial Grippers: State-of-the-Art and Main Design Characteristics, in: 2013: pp. 107–131. https://doi.org/10.1007/978-1-4471-4664-3_5.
- [121] G.C. J. Cuadrado, M.A. Naya, M. Ceccarelli, On the design of driving mechanisms in two-finger grippers, *Proc. Int. Symp. Ind. Robot.* 23 (1992) 568–568.
- [122] B.K. A. K. Jaiswal, Material Compositions Affects Vacuum System in Gripping Technology, *Int. J. Sci. Technol.* 7 (2017).
- [123] Y. Chen, S. Guo, C. Li, H. Yang, L. Hao, Size recognition and adaptive grasping using an integration of actuating and sensing soft pneumatic gripper, *Rob. Auton. Syst.* 104 (2018) 14–24. <https://doi.org/10.1016/j.robot.2018.02.020>.
- [124] C. Lanni, M. Ceccarelli, An Optimization Problem Algorithm for Kinematic Design of Mechanisms for Two-Finger Grippers, *Open Mech. Eng. J.* 3 (2009) 49–62. <https://doi.org/10.2174/1874155X00903010049>.
- [125] L. Silva-Caballero, A., González-Palacios, M., Aguilera-Cortés, Implementation of the Slide-O-Cam mechanism in the design of a robot gripper, in: 13th World Congr. Mech. Mach. Sci., 2011.
- [126] N. Modler, W. Hufenbach, A. Comsa, I. Maniu, M. Zichner, J. Friedrich, Compliant Structures in Book Handling Applications, *Appl. Mech. Mater.* 162 (2012) 543–548. <https://doi.org/10.4028/www.scientific.net/AMM.162.543>.
- [127] J. Vollm, Prescription filling apparatus implementing a pick and place method, 2004.
- [128] G. Fantoni, D. Gabelloni, J. Tilli, Concept design of new grippers using abstraction and analogy, *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* 227 (2013) 1521–1532. <https://doi.org/10.1177/0954405413489967>.
- [129] G. Carbone, ed., *Grasping in Robotics*, Springer London, London, 2013. <https://doi.org/10.1007/978-1-4471-4664-3>.
- [130] A. Morales, M. Prats, J. Felip, Sensors and Methods for the Evaluation of Grasping, in: 2013: pp. 77–104. https://doi.org/10.1007/978-1-4471-4664-3_4.
- [131] F. Liu, W. Ji, W. Tang, B. Xu, D. Zhao, Research on Grasping Planning for Apple Picking Robot's End-Effector, in: 2016: pp. 95–103. https://doi.org/10.1007/978-3-662-48386-2_11.
- [132] J. Seliger, G.; Stephan, Flexible garment handling with adaptive control strategies, 29th Symp. Ind. Robot. ISR. (1998).
- [133] A. Parness, N. Abcouwer, C. Fuller, N. Wiltsie, J. Nash, B. Kennedy, LEMUR 3: A limbed climbing robot for extreme terrain mobility in space, in: 2017 IEEE Int. Conf. Robot. Autom., IEEE, 2017: pp. 5467–5473. <https://doi.org/10.1109/ICRA.2017.7989643>.
- [134] T. Ota, K. Hirata, T. Yamaguchi, Y. Kawase, K. Watanabe, A. Nakase, Dynamic response analysis of opening and closing sensor for windows, *IEEE Trans. Magn.* 41 (2005) 1604–1607. <https://doi.org/10.1109/TMAG.2005.849183>.
- [135] P. Kampmann, T. Stoffregen, F. Kirchner, Equipping industrial deep-sea manipulators with a sense of touch, in: *Ocean. 2015 - MTS/IEEE Washingt.*, IEEE, 2015: pp. 1–5. <https://doi.org/10.23919/OCEANS.2015.7401919>.
- [136] J. Shaw, K.Y. Cheng, Object identification and 3-D position calculation using eye-in-hand single camera for robot gripper, in: 2016 IEEE Int. Conf. Ind. Technol., IEEE, 2016: pp. 1622–1625. <https://doi.org/10.1109/ICIT.2016.7475004>.
- [137] C.-H. Liu, C.-H. Chiu, Design and prototype of monolithic compliant grippers for adaptive grasping,

- in: 2018 3rd Int. Conf. Control Robot. Eng., IEEE, 2018: pp. 51–55.
<https://doi.org/10.1109/ICCRE.2018.8376432>.
- [138] X. Jinyu, Z. Tiedong, H. Hai, Z. Zexing, T. Qirong, The development on a streamlined underwater vehicle for autonomous manipulation, in: Ocean. 2017 - Aberdeen, IEEE, 2017: pp. 1–6.
<https://doi.org/10.1109/OCEANSE.2017.8084612>.
 - [139] R.S. Peres, X. Jia, J. Lee, K. Sun, A.W. Colombo, J. Barata, Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook, IEEE Access. 8 (2020) 220121–220139.
<https://doi.org/10.1109/ACCESS.2020.3042874>.
 - [140] L. Morgenstern, S.A. McIlraith, John McCarthy's legacy, Artif. Intell. 175 (2011) 1–24.
<https://doi.org/10.1016/j.artint.2010.11.003>.
 - [141] A.M. Turing, I.—Computing Machinery And Intelligence, Mind. LIX (1950) 433–460.
<https://doi.org/10.1093/mind/LIX.236.433>.
 - [142] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133. <https://doi.org/10.1007/BF02478259>.
 - [143] John McCarthy, Programs With Common Sense, in: Proc. Teddington Conf. Mech. Thought Process., 1959: pp. 75–91.
 - [144] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain., Psychol. Rev. 65 (1958) 386–408. <https://doi.org/10.1037/h0042519>.
 - [145] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern. 36 (1980) 193–202.
<https://doi.org/10.1007/BF00344251>.
 - [146] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature. 323 (1986) 533–536. <https://doi.org/10.1038/323533a0>.
 - [147] J. Elman, Finding structure in time, Cogn. Sci. 14 (1990) 179–211. [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E).
 - [148] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE. 86 (1998) 2278–2324. <https://doi.org/10.1109/5.726791>.
 - [149] G.E. Hinton, S. Osindero, Y.-W. Teh, A Fast Learning Algorithm for Deep Belief Nets, Neural Comput. 18 (2006) 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
 - [150] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2009: pp. 248–255.
<https://doi.org/10.1109/CVPR.2009.5206848>.
 - [151] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM. 60 (2012) 84–90. <https://doi.org/10.1145/3065386>.
 - [152] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, (2014). <http://arxiv.org/abs/1406.2661>.
 - [153] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature. 529 (2016) 484–489.
<https://doi.org/10.1038/nature16961>.
 - [154] S. Sabour, N. Frosst, G.E. Hinton, Dynamic Routing Between Capsules, (2017).
<http://arxiv.org/abs/1710.09829>.
 - [155] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, (2020). <http://arxiv.org/abs/2005.14165>.
 - [156] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, in: 2017 IEEE Int. Conf. Comput. Vis., IEEE, 2017: pp. 2242–2251. <https://doi.org/10.1109/ICCV.2017.244>.
 - [157] J.S. Dan C. Ciresan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, Flexible, High Performance Convolutional Neural Networks for Image Classification, in: Proc. TWENTY-SECOND Int. Jt. Conf. Artif. Intell., 2011. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>.

- [158] D.C. Cireşan, U. Meier, L.M. Gambardella, J. Schmidhuber, Deep, Big, Simple Neural Nets for Handwritten Digit Recognition, *Neural Comput.* 22 (2010) 3207–3220. https://doi.org/10.1162/NECO_a_00052.
- [159] D.C. Cireşan, U. Meier, L.M. Gambardella, J. Schmidhuber, Convolutional Neural Network Committees for Handwritten Character Classification, in: 2011 Int. Conf. Doc. Anal. Recognit., IEEE, 2011: pp. 1135–1139. <https://doi.org/10.1109/ICDAR.2011.229>.
- [160] Y. Bengio, Learning Deep Architectures for AI, *Found. Trends® Mach. Learn.* 2 (2009) 1–127. <https://doi.org/10.1561/22000000006>.
- [161] R. Silwal, A. Alsadoon, P.W.C. Prasad, O.H. Alsadoon, A. Al-Qaraghuli, A novel deep learning system for facial feature extraction by fusing CNN and MB-LBP and using enhanced loss function, *Multimed. Tools Appl.* 79 (2020) 31027–31047. <https://doi.org/10.1007/s11042-020-09559-1>.
- [162] F. Zhi-Peng, Z. Yan-Ning, H. Hai-Yan, Survey of deep learning in face recognition, in: 2014 Int. Conf. Orange Technol., IEEE, 2014: pp. 5–8. <https://doi.org/10.1109/ICOT.2014.6954663>.
- [163] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S.Z. Li, T. Hospedales, When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition, in: 2015 IEEE Int. Conf. Comput. Vis. Work., IEEE, 2015: pp. 384–392. <https://doi.org/10.1109/ICCVW.2015.58>.
- [164] W. Liu, L. Zhou, J. Chen, Face Recognition Based on Lightweight Convolutional Neural Networks, *Information.* 12 (2021) 191. <https://doi.org/10.3390/info12050191>.
- [165] M.D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, in: 2014: pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53.
- [166] Y. LeCun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proc. 2010 IEEE Int. Symp. Circuits Syst., IEEE, 2010: pp. 253–256. <https://doi.org/10.1109/ISCAS.2010.5537907>.
- [167] P. Sermanet, Y. LeCun, Traffic sign recognition with multi-scale Convolutional Networks, in: 2011 Int. Jt. Conf. Neural Networks, IEEE, 2011: pp. 2809–2813. <https://doi.org/10.1109/IJCNN.2011.6033589>.
- [168] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, (2014). <http://arxiv.org/abs/1409.1556>.
- [169] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2015: pp. 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>.
- [170] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2016: pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- [171] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, in: 2017 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2017: pp. 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>.
- [172] H. Lee, Y. Largman, P. Pham, A.Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, in: 23rd Annu. Conf. Neural Inf. Process. Syst., 2009.
- [173] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proc. 26th Annu. Int. Conf. Mach. Learn. - ICML '09, ACM Press, New York, New York, USA, 2009: pp. 1–8. <https://doi.org/10.1145/1553374.1553453>.
- [174] J.M. Johnson, T.M. Khoshgoftaar, Survey on deep learning with class imbalance, *J. Big Data.* 6 (2019) 27. <https://doi.org/10.1186/s40537-019-0192-5>.
- [175] Kevin Gurney, An introduction to neural networks, UCL Press, 1997.
- [176] D. Duarte, N. Ståhl, Machine Learning: A Concise Overview, in: 2019: pp. 27–58. https://doi.org/10.1007/978-3-319-97556-6_3.
- [177] R. Cochocki, A. and Unbehauen, Neural Networks for Optimization and Signal Processing, 1st ed., John Wiley & Sons, Inc., 1993.
- [178] D.O. Hebb, The organization of behavior; a neuropsychological theory, Wiley, 1949.
- [179] Frank Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, 1962.

- [180] G. Hernández, E. Zamora, H. Sossa, G. Téllez, F. Furlán, Hybrid neural networks for big data classification, *Neurocomputing*. 390 (2020) 327–340. <https://doi.org/10.1016/j.neucom.2019.08.095>.
- [181] Y. Wang, Y. Li, Y. Song, X. Rong, The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition, *Appl. Sci.* 10 (2020) 1897. <https://doi.org/10.3390/app10051897>.
- [182] B. Ding, H. Qian, J. Zhou, Activation functions and their characteristics in deep neural networks, in: 2018 Chinese Control Decis. Conf., IEEE, 2018: pp. 1836–1841. <https://doi.org/10.1109/CCDC.2018.8407425>.
- [183] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), (2015). <http://arxiv.org/abs/1511.07289>.
- [184] A.Y.N. Andrew L. Maas, Awni Y. Hannun, Rectifier Nonlinearities Improve Neural Network Acoustic Models, in: Proc. 30 Th Int. Conf. Ma- Chine Learn., 2013.
- [185] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in: 2015 IEEE Int. Conf. Comput. Vis., IEEE, 2015: pp. 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>.
- [186] T. Szandała, Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks, in: 2021: pp. 203–224. https://doi.org/10.1007/978-981-15-5495-7_11.
- [187] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation Functions: Comparison of trends in Practice and Research for Deep Learning, (2018). <http://arxiv.org/abs/1811.03378>.
- [188] A. Goodfellow, Ian; Bengio, Yoshua; Courville, Deep Learning, MIT Press, 2016.
- [189] Hecht-Nielsen, Theory of the backpropagation neural network, in: Int. Jt. Conf. Neural Networks, IEEE, 1989: pp. 593–605 vol.1. <https://doi.org/10.1109/IJCNN.1989.118638>.
- [190] Q. Wang, Y. Ma, K. Zhao, Y. Tian, A Comprehensive Survey of Loss Functions in Machine Learning, *Ann. Data Sci.* 9 (2022) 187–212. <https://doi.org/10.1007/s40745-020-00253-5>.
- [191] K. Janocha, W.M. Czarnecki, On Loss Functions for Deep Neural Networks in Classification, (2017). <http://arxiv.org/abs/1702.05659>.
- [192] I.H. Sarker, Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions, *SN Comput. Sci.* 2 (2021) 420. <https://doi.org/10.1007/s42979-021-00815-1>.
- [193] T.S. Geoffrey E Hinton, Optimal perceptual inference, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 1983.
- [194] D. Ackley, G. Hinton, T. Sejnowski, A learning algorithm for boltzmann machines, *Cogn. Sci.* 9 (1985) 147–169. [https://doi.org/10.1016/S0364-0213\(85\)80012-4](https://doi.org/10.1016/S0364-0213(85)80012-4).
- [195] G.E. Hinton, R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, *Science* (80-.). 313 (2006) 504–507. <https://doi.org/10.1126/science.1127647>.
- [196] H.L. Ruslan Salakhutdinov, Efficient Learning of Deep Boltzmann Machines, in: Proc. Thirteen. Int. Conf. Artif. Intell. Stat., n.d.
- [197] G.E. Salakhutdinov, Ruslan; Hinton, Deep Boltzmann Machines, in: 12th Int. Confe-Rence Artif. Intell. Stat., 2009.
- [198] G. Montufar, Restricted Boltzmann Machines: Introduction and Review, (2018). <http://arxiv.org/abs/1806.07066>.
- [199] B. Ghogho, A. Ghodsi, F. Karray, M. Crowley, Restricted Boltzmann Machine and Deep Belief Network: Tutorial and Survey, (2021). <http://arxiv.org/abs/2107.12521>.
- [200] G.E. Hinton, Training Products of Experts by Minimizing Contrastive Divergence, *Neural Comput.* 14 (2002) 1771–1800. <https://doi.org/10.1162/089976602760128018>.
- [201] D.M.S. Arsa, G. Jati, A.J. Mantau, I. Wasito, Dimensionality reduction using deep belief network in big data case study: Hyperspectral image classification, in: 2016 Int. Work. Big Data Inf. Secur., IEEE, 2016: pp. 71–76. <https://doi.org/10.1109/IWBIS.2016.7872892>.
- [202] R. Sarikaya, G.E. Hinton, A. Deoras, Application of Deep Belief Networks for Natural Language Understanding, *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 22 (2014) 778–784. <https://doi.org/10.1109/TASLP.2014.2303296>.
- [203] Pierre Baldi, Autoencoders, Unsupervised Learning, and Deep Architectures, in: Proc. ICML Work.

Unsupervised Transf. Learn., 2012.

- [204] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction, in: 2011: pp. 52–59. https://doi.org/10.1007/978-3-642-21735-7_7.
- [205] A. Jabbar, X. Li, B. Omar, A Survey on Generative Adversarial Networks: Variants, Applications, and Training, *ACM Comput. Surv.* 54 (2022) 1–49. <https://doi.org/10.1145/3463475>.
- [206] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, H.-H. Dai, Recent Advances of Generative Adversarial Networks in Computer Vision, *IEEE Access.* 7 (2019) 14985–15006. <https://doi.org/10.1109/ACCESS.2018.2886814>.
- [207] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, M. Hasan, B.C. Van Essen, A.A.S. Awwal, V.K. Asari, A State-of-the-Art Survey on Deep Learning Theory and Architectures, *Electronics.* 8 (2019) 292. <https://doi.org/10.3390/electronics8030292>.
- [208] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, H. Anderson, H. Roff, G.C. Allen, J. Steinhardt, C. Flynn, S.Ó. Héigearthaigh, S. Beard, H. Belfield, S. Farquhar, C. Lyle, R. Crootof, O. Evans, M. Page, J. Bryson, R. Yampolskiy, D. Amodei, The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation, (2018). <http://arxiv.org/abs/1802.07228>.
- [209] D.H. Hubel, T.N. Wiesel, Receptive fields of single neurones in the cat's striate cortex, *J. Physiol.* 148 (1959) 574–591. <https://doi.org/10.1113/jphysiol.1959.sp006308>.
- [210] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J. Physiol.* 160 (1962) 106–154. <https://doi.org/10.1113/jphysiol.1962.sp006837>.
- [211] K. O'Shea, R. Nash, An Introduction to Convolutional Neural Networks, (2015). <http://arxiv.org/abs/1511.08458>.
- [212] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Comput.* 1 (1989) 541–551. <https://doi.org/10.1162/neco.1989.1.4.541>.
- [213] R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, *Insights Imaging.* 9 (2018) 611–629. <https://doi.org/10.1007/s13244-018-0639-9>.
- [214] Y. Yu, X. Si, C. Hu, J. Zhang, A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures, *Neural Comput.* 31 (2019) 1235–1270. https://doi.org/10.1162/neco_a_01199.
- [215] L. Alzubaidi, M.A. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, Robust application of new deep learning tools: an experimental study in medical imaging, *Multimed. Tools Appl.* 81 (2022) 13289–13317. <https://doi.org/10.1007/s11042-021-10942-9>.
- [216] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, S. Valaee, Recent Advances in Recurrent Neural Networks, (2017). <http://arxiv.org/abs/1801.01078>.
- [217] M.A. Kızrak, B. Bolat, Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma, *Bilişim Teknol. Derg.* (2018) 263–286. <https://doi.org/10.17671/gazibtd.419205>.
- [218] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Comput.* 9 (1997) 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [219] A. Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network, (2018). <https://doi.org/10.1016/j.physd.2019.132306>.
- [220] Francois Chollet, Deep Learning with Python, 2017.
- [221] P. Liashchynskiy, P. Liashchynskiy, Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS, (2019). <http://arxiv.org/abs/1912.06059>.
- [222] Y.B. James Bergstra, Random Search for Hyper-Parameter Optimization, *J. Mach. Learn. Res.* (2012) 281–305.
- [223] T. Agrawal, Hyperparameter Optimization in Machine Learning, 2021.
- [224] Nitish Srivastava; Geoffrey Hinton; Alex Krizhevsky; Ilya Sutskever; Ruslan Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15 (2014) 1929–1958.
- [225] S. Ruder, An overview of gradient descent optimization algorithms, (2016). <http://arxiv.org/abs/1609.04747>.
- [226] L. Luo, Y. Xiong, Y. Liu, X. Sun, Adaptive Gradient Methods with Dynamic Bound of Learning

- Rate, (2019). <http://arxiv.org/abs/1902.09843>.
- [227] X. Qian, D. Klabjan, The Impact of the Mini-batch Size on the Variance of Gradients in Stochastic Gradient Descent, (2020). <http://arxiv.org/abs/2004.13146>.
 - [228] M. Li, T. Zhang, Y. Chen, A.J. Smola, Efficient mini-batch training for stochastic optimization, in: Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., ACM, New York, NY, USA, 2014: pp. 661–670. <https://doi.org/10.1145/2623330.2623612>.
 - [229] A. Kulkarni, D. Chong, F.A. Batareseh, Foundations of data imbalance and solutions for a data democracy, in: Data Democr., Elsevier, 2020: pp. 83–106. <https://doi.org/10.1016/B978-0-12-818366-3.00005-8>.
 - [230] B. Behera, G. Kumaravelan, P. Kumar B., Performance Evaluation of Deep Learning Algorithms in Biomedical Document Classification, in: 2019 11th Int. Conf. Adv. Comput., IEEE, 2019: pp. 220–224. <https://doi.org/10.1109/ICoAC48765.2019.246843>.
 - [231] A. Parvat, J. Chavan, S. Kadam, S. Dev, V. Pathak, A survey of deep-learning frameworks, in: 2017 Int. Conf. Inven. Syst. Control, IEEE, 2017: pp. 1–7. <https://doi.org/10.1109/ICISC.2017.8068684>.
 - [232] K. Collobert, Ronan; Kavukcuoglu, Torch7: A matlab-like environment for machine learning, in: NIPS Work., 2011.
 - [233] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, Y. Bengio, Theano: new features and speed improvements, (2012). <http://arxiv.org/abs/1211.5590>.
 - [234] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe, in: Proc. 22nd ACM Int. Conf. Multimed., ACM, New York, NY, USA, 2014: pp. 675–678. <https://doi.org/10.1145/2647868.2654889>.
 - [235] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, (2016). <http://arxiv.org/abs/1603.04467>.
 - [236] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, Á. López García, I. Heredia, P. Malík, L. Hluchý, Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey, Artif. Intell. Rev. 52 (2019) 77–124. <https://doi.org/10.1007/s10462-018-09679-z>.
 - [237] A. Vedaldi, K. Lenc, MatConvNet - Convolutional Neural Networks for MATLAB, (2014). <http://arxiv.org/abs/1412.4564>.
 - [238] R. Bellman, A Markovian Decision Process, J. Math. Mech. 6 (1957) 679–684. <https://doi.org/http://www.jstor.org/stable/24900506>.
 - [239] R. Bellman, Dynamic programming and stochastic control processes, Inf. Control. 1 (1958) 228–239. [https://doi.org/10.1016/S0019-9958\(58\)80003-0](https://doi.org/10.1016/S0019-9958(58)80003-0).
 - [240] D.P. Bertsekas, Dynamic Programming and Stochastic Control, Academic Press, 2001.
 - [241] M.L. Puterman, Markov Decision Processes, Wiley, 1994. <https://doi.org/10.1002/9780470316887>.
 - [242] E. V. Denardo, Dynamic Programming: Models and Applications, Dover Publications, 2003.
 - [243] F.B. Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.
 - [244] A.G. Barto, R.S. Sutton, C.W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Trans. Syst. Man. Cybern. SMC-13 (1983) 834–846. <https://doi.org/10.1109/TSMC.1983.6313077>.
 - [245] B. Jang, M. Kim, G. Harerimana, J.W. Kim, Q-Learning Algorithms: A Comprehensive Classification and Applications, IEEE Access. 7 (2019) 133653–133667. <https://doi.org/10.1109/ACCESS.2019.2941229>.
 - [246] C.J.C.H. Watkins, Learning from delayed rewards, University of Cambridge, 1989.
 - [247] C.J.C.H. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292. <https://doi.org/10.1007/BF00992698>.
 - [248] G. Tesauro, Temporal difference learning and TD-Gammon, Commun. ACM. 38 (1995) 58–68. <https://doi.org/10.1145/203330.203343>.

- [249] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning, (2013). <http://arxiv.org/abs/1312.5602>.
- [250] J. Schulman, S. Levine, P. Moritz, M.I. Jordan, P. Abbeel, Trust Region Policy Optimization, (2015). <http://arxiv.org/abs/1502.05477>.
- [251] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, (2017). <http://arxiv.org/abs/1707.06347>.
- [252] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, (2015). <http://arxiv.org/abs/1509.02971>.
- [253] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, (2018). <http://arxiv.org/abs/1801.01290>.
- [254] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, *Nature*. 518 (2015) 529–533. <https://doi.org/10.1038/nature14236>.
- [255] M. Gregurić, M. Vujić, C. Alexopoulos, M. Miletić, Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data, *Appl. Sci.* 10 (2020) 4011. <https://doi.org/10.3390/app10114011>.
- [256] H. Van Hasselt, A. Guez, D. Silver, Deep Reinforcement Learning with Double Q-Learning, *Proc. AAAI Conf. Artif. Intell.* 30 (2016). <https://doi.org/10.1609/aaai.v30i1.10295>.
- [257] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, N. de Freitas, Dueling Network Architectures for Deep Reinforcement Learning, (2015). <http://arxiv.org/abs/1511.06581>.
- [258] A. Phinyomark, E. Scheme, EMG Pattern Recognition in the Era of Big Data and Deep Learning, *Big Data Cogn. Comput.* 2 (2018) 21. <https://doi.org/10.3390/bdcc2030021>.
- [259] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*. 61 (2015) 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>.
- [260] A. Shrestha, A. Mahmood, Review of Deep Learning Algorithms and Architectures, *IEEE Access*. 7 (2019) 53040–53065. <https://doi.org/10.1109/ACCESS.2019.2912200>.
- [261] Y.N. Singh, S.K. Singh, A.K. Ray, Bioelectrical Signals as Emerging Biometrics: Issues and Challenges, *ISRN Signal Process.* 2012 (2012) 1–13. <https://doi.org/10.5402/2012/712032>.
- [262] J. Qi, G. Jiang, G. Li, Y. Sun, B. Tao, Surface EMG hand gesture recognition system based on PCA and GRNN, *Neural Comput. Appl.* 32 (2020) 6343–6351. <https://doi.org/10.1007/s00521-019-04142-8>.
- [263] A. Pal, A.K. Gautam, Y.N. Singh, Evaluation of Bioelectric Signals for Human Recognition, *Procedia Comput. Sci.* 48 (2015) 746–752. <https://doi.org/10.1016/j.procs.2015.04.211>.
- [264] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U. Rajendra Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Comput. Biol. Med.* 121 (2020) 103792. <https://doi.org/10.1016/j.compbiomed.2020.103792>.
- [265] F. Murat, O. Yildirim, M. Talo, U.B. Baloglu, Y. Demir, U.R. Acharya, Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review, *Comput. Biol. Med.* 120 (2020) 103726. <https://doi.org/10.1016/j.compbiomed.2020.103726>.
- [266] Y. Celik, M. Talo, O. Yildirim, M. Karabatak, U.R. Acharya, Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images, *Pattern Recognit. Lett.* 133 (2020) 232–239. <https://doi.org/10.1016/j.patrec.2020.03.011>.
- [267] M. Talo, O. Yildirim, U.B. Baloglu, G. Aydin, U.R. Acharya, Convolutional neural networks for multi-class brain disease detection using MRI images, *Comput. Med. Imaging Graph.* 78 (2019) 101673. <https://doi.org/10.1016/j.compmedimag.2019.101673>.
- [268] V. Ruonala, E. Pekkonen, S. Rissanen, O. Airaksinen, G. Miroshnichenko, M. Kankaanpää, P. Karjalainen, Dynamic tension EMG to characterize the effects of DBS treatment of advanced Parkinson's disease, in: 2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., IEEE, 2014: pp. 3248–3251. <https://doi.org/10.1109/EMBC.2014.6944315>.
- [269] G. Gini, M. Arveti, I. Somlai, M. Folgheraiter, Acquisition and Analysis of EMG Signals to Recognize Multiple Hand Movements for Prosthetic Applications, *Appl. Bionics Biomech.* 9 (2012).
- [270] B. Zhang, S. Zhang, Pattern-based grasping force estimation from surface electromyography, in: 2017 Int. Conf. Algorithms, Methodol. Model. Appl. Emerg. Technol., IEEE, 2017: pp. 1–5.

- <https://doi.org/10.1109/ICAMMAET.2017.8186630>.
- [271] M.I.I. and O.O.K. Md. R. Ahsan, Advances in Electromyogram Signal Classification to Improve the Quality of Life for the Disabled and Aged People, *J. Comput. Sci.* 6 (2010) 706–715. <https://doi.org/10.3844/jcssp.2010.706.715>.
 - [272] D. Carroll, S. Alwarappan, Recent Advances in Biosensors and Biosensing Protocols, *J. Biosens. Bioelectron.* 03 (2012). <https://doi.org/10.4172/2155-6210.1000e112>.
 - [273] O. Iqbal, S.A. Fattah, S. Zahin, Hand movement recognition based on singular value decomposition of surface EMG signal, in: 2017 IEEE Reg. 10 Humanit. Technol. Conf., IEEE, 2017: pp. 837–842. <https://doi.org/10.1109/R10-HTC.2017.8289085>.
 - [274] S. Sengupta, S. Basak, P. Saikia, S. Paul, V. Tsalavoutis, F. Atiah, V. Ravi, A. Peters, A review of deep learning with special emphasis on architectures, applications and recent trends, *Knowledge-Based Syst.* 194 (2020) 105596. <https://doi.org/10.1016/j.knosys.2020.105596>.
 - [275] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare, *Nat. Med.* 25 (2019) 24–29. <https://doi.org/10.1038/s41591-018-0316-z>.
 - [276] H. Wang, N. Liu, Y. Zhang, D. Feng, F. Huang, D. Li, Y. Zhang, Deep reinforcement learning: a survey, *Front. Inf. Technol. Electron. Eng.* 21 (2020) 1726–1744. <https://doi.org/10.1631/FITEE.1900533>.
 - [277] T. Kaluarachchi, A. Reis, S. Nanayakkara, A Review of Recent Deep Learning Approaches in Human-Centered Machine Learning, *Sensors.* 21 (2021) 2514. <https://doi.org/10.3390/s21072514>.
 - [278] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, M. Dehmer, An Introductory Review of Deep Learning for Prediction Models With Big Data, *Front. Artif. Intell.* 3 (2020). <https://doi.org/10.3389/frai.2020.00004>.
 - [279] I. Arel, D.C. Rose, T.P. Karnowski, Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier], *IEEE Comput. Intell. Mag.* 5 (2010) 13–18. <https://doi.org/10.1109/MCI.2010.938364>.
 - [280] P. Padmanabhan, S. Puthusserypady, Nonlinear analysis of EMG signals - a chaotic approach, in: 26th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., IEEE, n.d.: pp. 608–611. <https://doi.org/10.1109/IEMBS.2004.1403231>.
 - [281] E. Scheme, K. Englehart, Training Strategies for Mitigating the Effect of Proportional Control on Classification in Pattern Recognition–Based Myoelectric Control, *JPO J. Prosthetics Orthot.* 25 (2013) 76–83. <https://doi.org/10.1097/JPO.0b013e318289950b>.
 - [282] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, Y. Laurillau, EMG feature evaluation for improving myoelectric pattern recognition robustness, *Expert Syst. Appl.* 40 (2013) 4832–4840. <https://doi.org/10.1016/j.eswa.2013.02.023>.
 - [283] R.N. Khushaba, A. Al-Timemy, S. Kodagoda, K. Nazarpour, Combined influence of forearm orientation and muscular contraction on EMG pattern recognition, *Expert Syst. Appl.* 61 (2016) 154–161. <https://doi.org/10.1016/j.eswa.2016.05.031>.
 - [284] A.H. Al-Timemy, R.N. Khushaba, G. Bugmann, J. Escudero, Improving the Performance Against Force Variation of EMG Controlled Multifunctional Upper-Limb Prostheses for Transradial Amputees, *IEEE Trans. Neural Syst. Rehabil. Eng.* 24 (2016) 650–661. <https://doi.org/10.1109/TNSRE.2015.2445634>.
 - [285] J. Qi, G. Jiang, G. Li, Y. Sun, B. Tao, Intelligent Human-Computer Interaction Based on Surface EMG Gesture Recognition, *IEEE Access.* 7 (2019) 61378–61387. <https://doi.org/10.1109/ACCESS.2019.2914728>.
 - [286] G. Li, H. Wu, G. Jiang, S. Xu, H. Liu, Dynamic Gesture Recognition in the Internet of Things, *IEEE Access.* 7 (2019) 23713–23724. <https://doi.org/10.1109/ACCESS.2018.2887223>.
 - [287] E. Scheme, K. Englehart, Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use, *J. Rehabil. Res. Dev.* 48 (2011) 643. <https://doi.org/10.1682/JRRD.2010.09.0177>.
 - [288] C. Sapsanis, G. Georgoulas, A. Tzes, D. Lymberopoulos, Improving EMG based classification of basic hand movements using EMD, in: 2013 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., IEEE, 2013: pp. 5754–5757. <https://doi.org/10.1109/EMBC.2013.6610858>.
 - [289] S. Bitzer, P. van der Smagt, Learning EMG control of a robotic hand: towards active prostheses, in:

- Proc. 2006 IEEE Int. Conf. Robot. Autom. 2006. ICRA 2006., IEEE, n.d.: pp. 2819–2823. <https://doi.org/10.1109/ROBOT.2006.1642128>.
- [290] S. Kim, J. Kim, M. Kim, S. Kim, J. Park, Grasping Force Estimation by sEMG Signals and Arm Posture: Tensor Decomposition Approach, *J. Bionic Eng.* 16 (2019) 455–467. <https://doi.org/10.1007/s42235-019-0037-0>.
 - [291] K. Li, Y. Fang, Y. Zhou, H. Liu, Non-Invasive Stimulation-Based Tactile Sensation for Upper-Extremity Prosthesis: A Review, *IEEE Sens. J.* 17 (2017) 2625–2635. <https://doi.org/10.1109/JSEN.2017.2674965>.
 - [292] G. Li, L. Zhang, Y. Sun, J. Kong, Towards the sEMG hand: internet of things sensors and haptic feedback application, *Multimed. Tools Appl.* 78 (2019) 29765–29782. <https://doi.org/10.1007/s11042-018-6293-x>.
 - [293] A. Kayabasi, B. Yildiz, M.F. Aslan, A. Durdu, Comparison of ELM and ANN on EMG Signals Obtained for Control of Robotic-Hand, in: 2018 10th Int. Conf. Electron. Comput. Artif. Intell., IEEE, 2018: pp. 1–5. <https://doi.org/10.1109/ECAI.2018.8679074>.
 - [294] G. Ouyang, X. Zhu, Z. Ju, H. Liu, Dynamical Characteristics of Surface EMG Signals of Hand Grasps via Recurrence Plot, *IEEE J. Biomed. Heal. Informatics.* 18 (2014) 257–265. <https://doi.org/10.1109/JBHI.2013.2261311>.
 - [295] Z. Ju, H. Liu, Human Hand Motion Analysis With Multisensory Information, *IEEE/ASME Trans. Mechatronics.* 19 (2014) 456–466. <https://doi.org/10.1109/TMECH.2013.2240312>.
 - [296] N.M. Kakoty, S.M. Hazarika, Recognition of grasp types through principal components of DWT based EMG features, in: 2011 IEEE Int. Conf. Rehabil. Robot., IEEE, 2011: pp. 1–6. <https://doi.org/10.1109/ICORR.2011.5975398>.
 - [297] M. Khezri, M. Jahed, Surface Electromyogram signal estimation based on wavelet thresholding technique, in: 2008 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., IEEE, 2008: pp. 4752–4755. <https://doi.org/10.1109/IEMBS.2008.4650275>.
 - [298] Selahaddin Batuhan Akben, Low-cost and easy-to-use grasp classification, using a simple 2-channel surface electromyography (sEMG), *Biomed. Res.* 28 (2017).
 - [299] Y. Ruangpaisarn, S. Jaiyen, sEMG signal classification using SMO algorithm and singular value decomposition, in: 2015 7th Int. Conf. Inf. Technol. Electr. Eng., IEEE, 2015: pp. 46–50. <https://doi.org/10.1109/ICITEED.2015.7408910>.
 - [300] A. Nishad, A. Upadhyay, R.B. Pachori, U.R. Acharya, Automated classification of hand movements using tunable-Q wavelet transform based filter-bank with surface electromyogram signals, *Futur. Gener. Comput. Syst.* 93 (2019) 96–110. <https://doi.org/10.1016/j.future.2018.10.005>.
 - [301] C. Sapsanis, Recognition of Basic Hand Movements Using Electromyography, 2013.
 - [302] Z.-H. Zhou, J. Feng, Deep forest, *Natl. Sci. Rev.* 6 (2019) 74–86. <https://doi.org/10.1093/nsr/nwy108>.
 - [303] D.C. Toledo-Perez, J. Rodriguez-Resendiz, R.A. Gomez-Loenzo, A Study of Computing Zero Crossing Methods and an Improved Proposal for EMG Signals, *IEEE Access.* 8 (2020) 8783–8790. <https://doi.org/10.1109/ACCESS.2020.2964678>.
 - [304] T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.* 27 (2006) 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>.
 - [305] W.N. Zhu W, Zeng N, Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS® Implementations, *NESUG Proc. Heal. Care Life Sci.* 19 (2010).
 - [306] Ö. Yıldırım, U.B. Baloglu, U.R. Acharya, A deep convolutional neural network model for automated identification of abnormal EEG signals, *Neural Comput. Appl.* 32 (2020) 15857–15868. <https://doi.org/10.1007/s00521-018-3889-z>.
 - [307] Aurélien Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, 2017.
 - [308] J. Togelius, AI Researchers, Video Games Are Your Friends!, (2016). https://doi.org/10.1007/978-3-319-48506-5_1.
 - [309] J. Kahn, The Smartest Machines Are Playing Games, *Bloom. Businessweek.* (2017) 34–35.
 - [310] Siyuan Xu, History of AI design in video games and its development in RTS games - Final Step, (2018). https://sites.google.com/site/myangelcafe/articles/history_ai.
 - [311] P.C. Chris Fairclough, Michael Fagan, Brian Mac Namee, Research Directions for AI in Computer Games, 2005.

- [312] J. Hruska, The Quest to Improve Video Game AI, (2016) 16–20.
- [313] A. Nareyek, AI in Computer Games, *Queue*. 1 (2004) 58–65. <https://doi.org/10.1145/971564.971593>.
- [314] P.B.S. Serafim, Y.L.B. Nogueira, C.A. Vidal, J.B.C. Neto, Towards Playing a 3D First-Person Shooter Game Using a Classification Deep Neural Network Architecture, in: 2017 19th Symp. Virtual Augment. Real., IEEE, 2017: pp. 120–126. <https://doi.org/10.1109/SVR.2017.24>.
- [315] Parry Olson, The Warcraft Method: Why Google’s DeepMind Is Training AI In Virtual Worlds, *Forbes*. (2010).
- [316] J.T. Georgios N. Yannakakis, Artificial Intelligence and Games, 2018.
- [317] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, D. Hassabis, Mastering the game of Go without human knowledge, *Nature*. 550 (2017) 354–359. <https://doi.org/10.1038/nature24270>.
- [318] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.P. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous Methods for Deep Reinforcement Learning, (2016). <http://arxiv.org/abs/1602.01783>.
- [319] Christof Koch, How the Computer Beat the Go Player, *Sci. Am. Mind*. (2016).
- [320] Chris Stokel-Walker, DeepMind AI thrashes human professionals at video game StarCraft II, (2019).
- [321] H. Tsukayama, OpenAI’s bot beat a human at video games last year. Now it will take on five at once, *Washingt. Post*. (2018).
- [322] K. Quach, OpenAI bots thrash team of Dota 2 semi-pros, set eyes on mega-tourney, *Regist*. (2018).
- [323] The International 2018: Results, (2018). https://liquipedia.net/dota2/The_International/2018.
- [324] Zhongzhi Shi, Advanced Artificial Intelligence, 2011.
- [325] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, S. Levine, Deep Reinforcement Learning for Vision-Based Robotic Grasping: A Simulated Comparative Evaluation of Off-Policy Methods, in: 2018 IEEE Int. Conf. Robot. Autom., IEEE, 2018: pp. 6284–6291. <https://doi.org/10.1109/ICRA.2018.8461039>.
- [326] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, T. Funkhouser, Learning Synergies between Pushing and Grasping with Self-supervised Deep Reinforcement Learning, (2018). <http://arxiv.org/abs/1803.09956>.
- [327] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, K. Goldberg, Learning ambidextrous robot grasping policies, *Sci. Robot.* 4 (2019). <https://doi.org/10.1126/scirobotics.aau4984>.
- [328] S. Joshi, S. Kumra, F. Sahin, Robotic Grasping using Deep Reinforcement Learning, in: 2020 IEEE 16th Int. Conf. Autom. Sci. Eng., IEEE, 2020: pp. 1461–1466. <https://doi.org/10.1109/CASE48305.2020.9216986>.
- [329] M. Ciocarlie, K. Hsiao, E.G. Jones, S. Chitta, R.B. Rusu, I.A. Şucan, Towards Reliable Grasping and Manipulation in Household Environments, in: 2014: pp. 241–252. https://doi.org/10.1007/978-3-642-28572-1_17.
- [330] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, J. Xiao, Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge, (2016). <http://arxiv.org/abs/1609.09475>.
- [331] F.-J. Chu, R. Xu, P.A. Vela, Real-world Multi-object, Multi-grasp Detection, (2018). <http://arxiv.org/abs/1802.00520>.
- [332] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, N. Zheng, ROI-based Robotic Grasp Detection for Object Overlapping Scenes, in: 2019 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2019: pp. 4768–4775. <https://doi.org/10.1109/IROS40897.2019.8967869>.
- [333] E. Jang, C. Devin, V. Vanhoucke, S. Levine, Grasp2Vec: Learning Object Representations from Self-Supervised Grasping, (2018). <http://arxiv.org/abs/1811.06964>.
- [334] Y. Song, L. Gao, X. Li, W. Shen, A novel robotic grasp detection method based on region proposal networks, *Robot. Comput. Integr. Manuf.* 65 (2020) 101963. <https://doi.org/10.1016/j.rcim.2020.101963>.
- [335] M.A. Moussa, M.S. Kamel, A connectionist model for learning robotic grasps using reinforcement learning, in: Proc. Int. Conf. Neural Networks, IEEE, n.d.: pp. 1771–1776.

- <https://doi.org/10.1109/ICNN.1996.549169>.
- [336] S. Amarjyoti, Deep Reinforcement Learning for Robotic Manipulation-The state of the art, (2017). <http://arxiv.org/abs/1701.08878>.
 - [337] Y. Deng, X. Guo, Y. Wei, K. Lu, B. Fang, D. Guo, H. Liu, F. Sun, Deep Reinforcement Learning for Robotic Pushing and Picking in Cluttered Environment, in: 2019 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2019: pp. 619–626. <https://doi.org/10.1109/IROS40897.2019.8967899>.
 - [338] H. Zhang, J. Tang, S. Sun, X. Lan, Robotic Grasping from Classical to Modern: A Survey, (2022). <http://arxiv.org/abs/2202.03631>.
 - [339] M. Körber, J. Lange, S. Rediske, S. Steinmann, R. Glück, Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning, (2021). <http://arxiv.org/abs/2103.04616>.
 - [340] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, J. Nieto, Comparing Task Simplifications to Learn Closed-Loop Object Picking Using Deep Reinforcement Learning, IEEE Robot. Autom. Lett. 4 (2019) 1549–1556. <https://doi.org/10.1109/LRA.2019.2896467>.



ÖZGEÇMİŞ

Musab COŞKUN

[Redacted text block]

[Redacted text block]

[Redacted text block]

[Redacted text block]

AKADEMİK FAALİYETLER

Makaleler:

1. Coskun, M., Yildirim, O., Demir, Y. *et al.*, “Efficient Deep Neural Network Model for Classification of Grasp Types Using sEMG Signals”, *J Ambient Intell Human Comput* (2021), <https://doi.org/10.1007/s12652-021-03284-9> (SCI).
2. Coşkun M., Yıldırım Ö., Uçar A., Demir Y., “An Overview of Popular Deep Learning Methods”, *European Journal of Technique (EJT)*, (2017), 7(2), 165-176.

Bildiriler:

1. M. Coşkun, A. Uçar, Ö. Yildirim and Y. Demir, “Face recognition based on convolutional neural network”, 2017 International Conference on Modern Electrical and Energy Systems (MEES), (2017), pp. 376-379, <https://doi.org/10.1109/MEES.2017.8248937>.
2. M. Coşkun, Ö. Yildirim and Y. Demir, “Robotic Grasping in Simulation Using Deep Reinforcement Learning”, 7th International Conference on Computer Science and Engineering, (2022).